

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE  
À L'OBTENTION DE LA  
MAÎTRISE EN GÉNIE ÉLECTRIQUE  
M. ING.

PAR  
JEAN-FRANÇOIS BOLAND

CONCEPTION D'UNE UNITÉ DE CONTRÔLE ET D'UN SYSTÈME DE  
POSITIONNEMENT POUR UN HÉLICOPTÈRE À PROPULSION HUMAINE

MONTREAL, FÉVRIER 2001

© droits réservés de Jean-François Boland

**CE MÉMOIRE A ÉTÉ ÉVALUÉ**

**PAR UN JURY COMPOSÉ DE:**

- **M. Jean Belzile, directeur de mémoire**  
Département de génie électrique à l'École de technologie supérieure
- **M. Fred Awad, professeur**  
Département de génie électrique à l'École de technologie supérieure
- **M. Louis Lamarche, professeur**  
Département de génie mécanique à l'École de technologie supérieure

**IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET UN PUBLIC**

**LE 29 JANVIER 2001**

**À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

# CONCEPTION D'UN MODULE DE CONTRÔLE ET D'UN SYSTÈME DE POSITIONNEMENT POUR UN HÉLICOPTÈRE À PROPULSION HUMAINE

Jean-François Boland

## Sommaire

Le design de l'hélicoptère à propulsion humaine Hélios repose sur la conception d'un module de contrôle et d'un système de positionnement capable d'assurer la stabilité de l'engin et de maximiser le transfert de puissance entre le pédaleur et les rotors. Pour accomplir ces tâches, un ordinateur de bord est intégré à l'appareil pour ajuster en temps réel l'angle d'attaque des pales. De cette façon il est possible d'obtenir un vecteur de poussée verticale, contribuant ainsi à la stabilité de l'engin. De plus, ce contrôle permet de maximiser en tout temps la puissance développée par le pédaleur. Un effort moins important est requis pour démarrer la rotation des rotors en diminuant le pas des pales. De cette façon il est possible de concentrer la puissance requise du pédaleur pour atteindre l'altitude de trois mètres.

Les ajustements des angles d'attaque des pales seront assurés par un contrôleur en boucle ouverte, s'appuyant sur une table de données préalablement calculée. Ces valeurs d'angle des pales seront en effet tabulées suite à des simulations sur le banc d'essais de l'assemblage pales/rotor. Le cœur du module de contrôle est constitué d'un microcontrôleur et de cartes d'entrées/sorties. Cette approche assure un minimum de poids ajouté à l'appareil tout en offrant la puissance de calcul requise.

D'autre part, même si l'hélicoptère est conçu pour être stable, plusieurs facteurs peuvent influencer sa position en vol. C'est pourquoi un système de positionnement utilisant la vision par ordinateur s'assurera de déterminer la position exacte de l'appareil en temps réel pour permettre d'apporter les corrections qui s'imposent. Cette méthode est précise et n'encombre pas inutilement l'aéronef, préservant ainsi sa légèreté. L'algorithme de traitement des images utilise Matlab comme plate-forme pour profiter de son traitement matriciel efficace. Toutefois, il est recommandé d'étudier la possibilité de transposer les algorithmes en langage C pour profiter de la vitesse d'exécution de ce langage.

## REMERCIEMENTS

Le projet Hélios est le fruit de la collaboration assidue de plusieurs personnes ayant un but commun: faire voler pour la première fois un hélicoptère à propulsion humain durant 60 secondes. Une passion du défi est partagée par les membres de l'équipe et cette énergie commune nous a permis de progresser constamment au cours des trois dernières années. Je ne pourrais passer sous silence le travail exceptionnel de Simon Joncas, chef de l'équipe. Autant sur le plan technique que sur le plan moral, Simon a toujours su encourager et motiver la troupe pour atteindre les objectifs visés.

Il m'est impossible de citer ici tous les membres de l'équipe, mais je tiens à souligner le travail remarquable accompli par Christian Belleau, Daniel Côté et Alexis Michaud. Ils ont su mener de front leurs projets respectifs et ce malgré les nombreuses difficultés de conception et de fabrication rencontrées depuis trois ans.

D'autre part, je remercie particulièrement M. Jean Belzile, directeur de thèse, qui m'a toujours appuyé et conseillé dans mes différentes démarches. Sur le plan professionnel et technique, Jean a su me guider et me renseigner adéquatement, me permettant ainsi de réaliser un projet de grande envergure. Une mention s'impose à ma

famille, amis et conjointe qui m'ont constamment encouragé et supporté tout au long de ce projet, malgré les nombreux sacrifices que cela imposait.

La fabrication de l'hélicoptère Hélios a été rendue possible grâce à la participation de nombreux commanditaires. C'est plus de 40 compagnies qui ont cru en notre idée et qui nous ont offert diverses commandites matérielles et monétaires. Un gros merci à ces entreprises qui participent activement au succès des projets étudiants de l'ÉTS.

## TABLE DES MATIÈRES

	Page
SOMMAIRE .....	i
REMERCIEMENTS .....	ii
LISTE DES TABLEAUX .....	vii
LISTE DES FIGURES .....	viii
LISTE DES ABRÉVIATIONS ET SIGLES .....	x
INTRODUCTION.....	1
CHAPITRE 1 MISE EN SITUATION .....	3
1.1 Le concours Igor Sikorsky .....	3
1.2 Historique des hélicoptères à propulsion humaine .....	4
1.3 Le projet Hélios.....	7
1.3.1 Choix de la configuration.....	8
1.3.1.1 Nombre de pédaleur .....	8
1.3.1.2 Disposition de la nacelle .....	10
1.3.2 Théorie de poussée des pales .....	13
1.3.2.1 Structure de la pale.....	13
1.3.2.2 Système à pas variable .....	13
1.3.3 Système de positionnement.....	14
1.4 Structure du mémoire.....	15
CHAPITRE 2 MODULE DE CONTRÔLE.....	16
2.1 Méthodologie de contrôle .....	16
2.1.1 Théorie de poussée des pales .....	17

2.1.2 Boucle de contrôle .....	19
2.1.2.1 Boucle fermée .....	20
2.1.2.2 Boucle ouverte .....	21
2.2 Unité de contrôle.....	22
2.2.1 Choix et description de l'ordinateur de bord .....	23
2.2.2 Disposition de l'unité de contrôle.....	24
2.2.3 Entrées-sorties du système .....	26
2.2.3.1 Carte de contrôle moteur.....	27
2.2.3.2 Altimètre .....	28
2.2.3.3 Encodeur .....	29
2.2.3.4 Carte de communication RF .....	29
2.2.3.5 Niveau mètre électronique .....	30
2.2.3.6 Gyroscope .....	31
2.3 Algorithme de vol .....	32
2.3.1 Contrôle de la rotation de la nacelle .....	33
2.4 Programmation.....	35
CHAPITRE 3 SIMULATEUR.....	37
3.1 Configuration du banc d'essais .....	38
3.1.1 Théorie du calcul du vecteur de poussée .....	40
3.2 Système de contrôle .....	41
3.3 Système d'acquisition .....	42
3.3.1 Choix de l'unité d'acquisition.....	43
3.3.2 Carte d'acquisition.....	45
3.3.3 Interface d'analyse de données .....	48
3.4 Procédure de calibrage .....	49
CHAPITRE 4 SYSTÈME DE POSITIONNEMENT .....	51
4.1 Paramètres de positionnement de l'HPH.....	51
4.2 Méthodes pour évaluer la position relative d'un objet .....	53

4.2.1 Positionnement avec intervention humaine .....	54
4.2.2 Global Positioning System.....	55
4.2.3 Accéléromètres .....	56
4.2.4 Vision par ordinateur .....	58
4.3 Configuration du système .....	60
4.4 Logiciel de vision "Héli-Tracker" .....	62
4.4.1 Choix du langage de programmation.....	64
4.4.2 Stratégie de développement .....	65
4.4.3 Module générateur d'images .....	67
4.4.4 Module détection de la cible .....	69
4.4.4.1 Conversion de l'image couleur en niveau de gris.....	69
4.4.4.2 Augmentation du contraste de l'image.....	70
4.4.4.3 Réduction du bruit .....	71
4.4.4.4 Conversion en image logique.....	72
4.4.4.5 Détection des objets et sélection de la cible.....	73
4.4.5 Module réduction de l'image.....	74
4.4.5.1 Réduction de la résolution versus réduction du champ de vision..	74
4.4.5.2 Implantation de l'algorithme .....	78
4.4.6 Évaluation de la position de la cible .....	81
4.4.7 Analyse des performances.....	81
4.5 Repositionnement de l'hélicoptère .....	83
DISCUSSION ET INTERPRÉTATION DES RÉSULTATS .....	85
CONCLUSION .....	88
RECOMMANDATIONS.....	90
BIBLIOGRAPHIE .....	92
ANNEXE A Règlements du défi Igor Sikorsky .....	94
ANNEXE B Code source programme Héli-Tracker .....	97
ANNEXE C Schémas électriques .....	109



## LISTE DES TABLEAUX

	Page
Tableau 1.1 Configurations étudiées .....	11
Tableau 2.1 Exemple de la table de contrôle du pas .....	32
Tableau 2.2 Segmentation du programme de contrôle en fichiers. ....	35
Tableau 4.1 Résumé des méthodes d'évaluation de la position .....	59
Tableau 4.2 Segmentation du programme Héli-Tracker .....	66
Tableau 4.3 Paramètres d'entrée du générateur d'images.....	68
Tableau 4.4 Temps d'exécution moyen des fonctions .....	82

## LISTE DES FIGURES

	Page
Figure 1.1 Hélicoptère à propulsion humaine DaVinci III (CalPoly) .....	5
Figure 1.2 Hélicoptère à propulsion humaine Yuri I (Nihon University) .....	6
Figure 1.3 Hélicoptère à propulsion humaine Yuri II, Nihon University.....	7
Figure 1.4 Banc d'essais pédaleur.....	9
Figure 1.5 Configuration de l'hélicoptère à propulsion humaine Hélios.....	12
Figure 2.1 Poussées générées par les pales .....	18
Figure 2.2 Angle d'attaque de la pale .....	19
Figure 2.3 Boucle de contrôle fermée .....	20
Figure 2.4 Boucle de contrôle ouverte .....	22
Figure 2.5 Ordinateur de bord Hélios-CPU .....	23
Figure 2.6 Position des moteurs et de l'ordinateur Hélios-CPU.....	25
Figure 2.7 Disposition des modules de commande pour les servomoteurs .....	25
Figure 2.8 Couplage électrique rotatif.....	26
Figure 2.9 Moteur Maxon assemblé sur le système à pas variable du rotor. ....	27
Figure 2.10 Mesure de l'altitude par télémétrie.....	29
Figure 2.11 Axes de détection de l'accéléromètre et signal de sortie.....	31
Figure 2.12 Ordinogramme du processus de contrôle.....	34
Figure 3.1 Plan mécanique du banc d'essais .....	38
Figure 3.2 Plateaux de cellules de charge .....	39
Figure 3.3 Disposition des cellules de charge sur le plateau triangulaire .....	40
Figure 3.4 Boucle de contrôle fermé du moteur hydraulique.....	42

Figure 3.5 Disposition du système d'acquisition.....	43
Figure 3.6 Diagramme bloc de l'unité d'acquisition et de contrôle .....	45
Figure 3.7 Pont de Wheatstone d'une cellule de charge.....	46
Figure 3.8 Étapes de conversion du signal d'une cellule de charge.....	47
Figure 3.9 Interface Pc pour l'analyse des données .....	48
Figure 4.1 Position de décolage de l'HPH.....	52
Figure 4.2 Diagramme bloc de l'accéléromètre ADXL200.....	57
Figure 4.3 Chronogramme de la sortie numérique de l'ADXL200 .....	57
Figure 4.4 Disposition des caméras.....	60
Figure 4.5 Images filtrées obtenues avec les caméras.....	61
Figure 4.6 Organigramme fonctionnel du système de positionnement.....	63
Figure 4.7 Segmentation du logiciel Héli-Tracker en plusieurs fichiers.....	65
Figure 4.8 Représentation matricielle d'une image .....	69
Figure 4.9 Conversion couleur à niveaux de gris.....	70
Figure 4.10 Augmentation du contraste de l'image.....	71
Figure 4.11 Filtrage de l'image avec un filtre médian.....	72
Figure 4.12 Conversion de l'image en image logique binaire .....	72
Figure 4.13 Cible détectée.....	74
Figure 4.14 Réduction de la résolution versus réduction du champ de vision.....	76
Figure 4.15 Exemple d'un champ de vision avec arrière plan du simulateur d'images....	77
Figure 4.16 Réduction du champ de vision et élimination des fausses cibles .....	77
Figure 4.17 Schéma bloc du zonage de l'image .....	78
Figure 4.18 Organigramme opérationnel de l'algorithme de poursuite de la cible .....	80
Figure 4.19 Calcul de la position de la cible dans la scène en trois dimensions.....	81
Figure 4.20 Image complexe et simple utilisées pour les tests. ....	83

## LISTE DES ABRÉVIATIONS ET SIGLES

HPH	Hélicoptère à propulsion humaine
APH	Avion à propulsion humaine
AHS	American Helicopter Society
IHPVA	International Human Powered Vehicle Association
RF	Radio fréquence
DAC	"Digital to Analog Converter"
ADC	"Analog to Digital Converter"
TPU	"Time Processing Unit"
QSPI	"Queue Serial Peripheral Interface"
CMOS	" Complementary Metal-Oxide Semiconductor"
P	Puissance, Watt
$\omega$	Vitesse de rotation angulaire, rad/s
T	Couple, N.m
RPM	Rotation par minute, r/min
$L_R$	Vecteur de poussée total d'un rotor, Kg
$L_B$	Vecteur de poussée d'une pale, Kg
$T_R$	Couple absorbé par le rotor, N.m
$\alpha$	Angle d'attaque d'une pale, °

## INTRODUCTION

Animés par la passion de relever des défis technologiques et par l'expérience acquise au cours de ces années, une équipe de quatre personnes décidèrent de s'attaquer à un nouveau challenge : la conception et la fabrication d'un hélicoptère à propulsion humaine (HPH). C'est ainsi que le projet Hélios a vu le jour en 1998 à l'École de Technologie Supérieure (ETS) pour tenter de relever le défi Igor Sikorsky (voir annexe A : Règlements du défi Sikorsky.)

La conception d'un hélicoptère à propulsion humaine est un projet d'envergure nécessitant la collaboration de chacun des membres de l'équipe autant du point de vue mécanique qu'électrique. Le design de l'HPH Hélios constitue le sujet du mémoire de maîtrise de trois étudiants en génie mécanique et un étudiant en génie électrique. Le mandat qui m'a été confié consiste à développer un module de contrôle ainsi qu'un système de positionnement pour cet hélicoptère.

Comme sa définition l'impose, un HPH est propulsé uniquement à l'aide de la puissance d'un athlète humain. Par conséquent, lors d'un vol, le pilote concentre toutes ses énergies dans le mouvement du pédalier pour défier les lois de la gravité. Ainsi, il lui est impossible d'exercer un quelconque contrôle sur l'engin ou même de déterminer sa position dans les airs. Il doit alors s'en remettre à l'ordinateur de bord qui s'acquittera des ces tâches complexes. Les critères du concours Igor Sikorsky imposent à l'HPH de maintenir sa position à l'intérieur d'un périmètre préétabli et ainsi réaliser un vol stationnaire. Il est par conséquent essentiel de détecter tout changement de position et d'assurer un parfait contrôle et une bonne stabilité en vol. Le module de contrôle intégré à l'HPH a comme première fonction de s'assurer de la stabilité en vol pour réduire au

minimum l'inclinaison de l'appareil et ainsi réduire les possibilités de déplacements de ce dernier. Cependant, différents facteurs, tel que le vent, peuvent affecter la position de l'HPH, c'est pourquoi un système de positionnement et un module de contrôle ont été conçus dans le but d'identifier rapidement la position de l'hélicoptère et d'apporter les corrections nécessaires au plan de vol et ce sans l'intervention du pilote.

Par ailleurs, étant donnée la faible puissance disponible pour propulser l'appareil, (moins de 750 Watts sur une période d'une minute), il devient capital de minimiser le poids total des composantes devant être intégrées à l'hélicoptère. Le module de contrôle sera alors constitué de circuits intégrés programmables permettant l'implantation d'un algorithme de vol et d'un programme de gestion pour la télémesure.

L'élaboration de ce système de positionnement et du module de contrôle en est à sa deuxième année de conception. En plus du design de modules électroniques devant prendre place à bord de l'hélicoptère, un banc d'essais complètement instrumenté fût fabriqué pour tester et développer l'algorithme de vol, ainsi que pour vérifier le comportement dynamique de l'HPH par l'intermédiaire de l'analyse des données recueillies.

## CHAPITRE 1

### MISE EN SITUATION

Le développement d'un système de contrôle et de positionnement pour un hélicoptère à propulsion humaine implique plusieurs choix de design qui relèvent directement de la configuration générale de l'appareil. Le présent chapitre présente une revue de la littérature des HPH avec laquelle nous avons été en mesure d'identifier la problématique de stabilité en vol et de déterminer une configuration optimale pour l'hélicoptère Hélios. La structure des prochains chapitres, où l'on retrouve l'essentiel de la méthodologie de développement de l'unité de contrôle et du système positionnement, est présentée à la fin du présent chapitre.

#### 1.1 Le concours Igor Sikorsky

Pour faire suite aux multiples challenges de conception d'avions à propulsion humaine et les très célèbres prix Kremer instaurés en 1959, l'American Helicopter Society (AHS) tente de récompenser depuis 1980 la première équipe à faire voler un hélicoptère à propulsion humaine (HPH). Ce vol historique s'inscrira dans le cadre d'un concours, nommé en l'honneur de l'inventeur de l'hélicoptère traditionnel, le « Igor Sikorsky Challenge » (voir annexe A : règlements du concours). Toutefois, pour remporter la bourse honorifique de 20 000\$ US, l'HPH doit effectuer un vol stationnaire de 60 secondes et atteindre momentanément une altitude de trois mètres. Durant toute la

durée du vol, l'appareil doit demeurer à l'intérieur d'une surface de 10 m<sup>2</sup> et la nacelle ne doit pas effectuer plus d'un tour sur elle-même. Au cours de ces 20 dernières années, plus d'une centaine de personnes ont investi temps, énergie et argent dans le but de relever ce défi d'ingénierie. Aucune équipe n'y est parvenue à ce jour, mais chacune a contribué à l'avancement des recherches dans le domaine de la propulsion humaine et laisse derrière elle un bagage de connaissances nouvelles.

## 1.2 Historique des hélicoptères à propulsion humaine

La puissance humaine a toujours fait l'objet de recherche dans le domaine des véhicules. Que ce soit sur la terre, sur la mer ou dans les airs, nombre d'équipes ont vu le jour depuis presque un siècle. L'organisme IHPVA (International Human Powered Vehicle Association) est une association qui fait la promotion de l'utilisation de la puissance humaine pour créer et innover dans le design d'engins à propulsion humaine. On retrouve sur leur site internet [10] un tableau d'honneur relatant les exploits réalisés dans les diverses catégories. Depuis 1933, les avions à propulsion humaine (APH) ont fait l'objet de plusieurs groupes de recherche et de nombreux prix et records. Beaucoup d'argent a été investi pour faire évoluer le tout premier design des Japonnais, le SUMPAC (1961), au modèle Daedalus de l'université MIT qui a parcouru une distance de 115.11 kilomètres en 1988.

Pour faire suite aux exploits réalisés avec les APH, plusieurs personnes se sont intéressées au monde des hélicoptères à propulsion humaine (HPH). Une vingtaine d'équipes, des quatre coins de la planète, ont vu le jour depuis 20 ans en plus de certains individus ayant volontairement contribué à la cause des hélicoptères à propulsion humaine. Toutefois, la complexité et l'envergure d'un tel projet n'ont pas permis à toutes ces équipes de voir leur prototype s'arracher de la gravité. En effet, depuis 1980, 17



HPH ont été fabriqués et seulement deux d'entre eux ont réussi à quitter le sol. Plusieurs essais ont bien sûr été réalisés avec les différents prototypes, mais aucun n'a satisfait aux exigences du concours.

Le premier vol officiel ayant été enregistré par l'AHS est celui de DaVinci III de la California Polytechnic University en 1989. Comme on peut le voir à la figure 1.1, l'hélicoptère est propulsé par deux hélices placées en bout de pôle, entraînées par des câbles reliés au pédalier. Cette technique a comme principal avantage d'éliminer le besoin d'un rotor de queue pour immobiliser l'appareil puisque aucun couple n'est transmis directement au rotor principal. L'équipe du professeur William B. Patterson a œuvré pendant plus de huit années à la conception de trois prototypes d'HPH : le DaVinci I (1984), le DaVinci II (1987) et le DaVinci III (1989). Les deux premiers engins n'ont jamais quitté le sol tandis que le dernier a réussi un vol de 7,1 secondes à seulement quelques centimètres du sol. L'instabilité de l'appareil l'empêcha de se maintenir plus longtemps dans les airs ou d'atteindre une altitude plus élevée. Il est à noter qu'aucun système de contrôle auxiliaire ne permettait à l'hélicoptère de maintenir une position stable. Le design des pâles à angle d'attaque fixe imposait donc comme hypothèse une stabilité continue tout le long du vol.

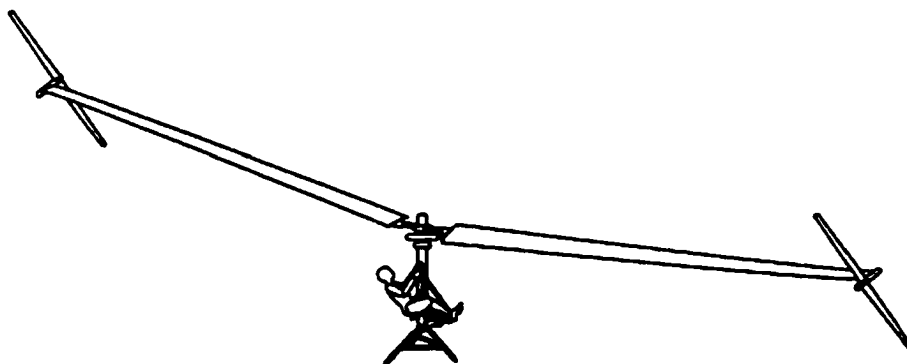


Figure 1.1 Hélicoptère à propulsion humaine DaVinci III (CalPoly)

Un peu plus de quatre années après le vol des Californiens, une équipe Japonaise de l'université Nihon, sous la supervision du professeur Akira Naito, tenta à son tour de réaliser ce vieux rêve de Léonard de Vinci avec leur prototype Yuri I. M. Naito n'en était pas à son premier concept d'HPH. Il a en effet développé, sur une période de neuf ans, quatre modèles d'hélicoptère à propulsion humaine: le Day Fly et le Papillon A et B et Yuri I. Seulement la quatrième génération a réussi à détrôner DaVinci III avec un vol d'une durée de 19,5 secondes à une altitude de 75 cm. Yuri I est propulsé par quatre rotors, deux tournants dans le sens horaire et les deux autres dans le sens anti-horaire pour annuler l'effet de rotation de la nacelle. Bien qu'encore loin de remporter le Challenge Igor Sikorsky, l'hélicoptère Yuri I a démontré une grande stabilité en vol. Cependant, l'HPH est dépourvu de système de positionnement et c'est ainsi qu'après 19,5 secondes l'appareil s'est retrouvé déporté hors de la surface de 100 mètres carrés. D'autres essais ont été réalisés avec ce prototype jusqu'au jour où, lors d'un vol, un bris dans la structure d'un des rotors mit fin à plusieurs années de travail minutieux et acharné. Jusqu'à ce jour, Yuri I détient toujours le record mondial de vol pour un HPH.

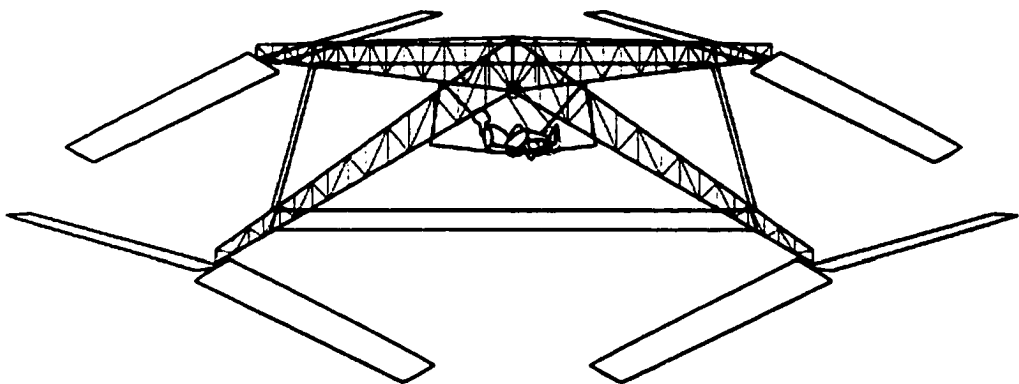


Figure 1.2 Hélicoptère à propulsion humaine Yuri I (Nihon University)

L'équipe japonaise n'est cependant pas prête de dire son dernier mot. Comme on peut le voir à la figure 1.3, un autre prototype, le Yuri II, est sur le point de prendre son

envol. Peu d'information est disponible à son sujet si ce n'est que cette photo qui nous montre clairement que la configuration du DaVinci III est de retour sur l'aire de décollage, mais cette fois avec trois pâles au lieu de deux.



Figure 1.3 Hélicoptère à propulsion humaine Yuri II, Nihon University

En conclusion, ces vingt années de recherche démontrent bien le niveau de difficulté de l'épreuve Sikorsky. Quelques équipes sont toujours actives dans le domaine nous laissant prévoir un avenir très prometteur d'autant plus que l'AHS a laissé entendre la possibilité de doubler la mise à 40 000\$ pour l'équipe victorieuse. Il est également connu que plusieurs travaux de recherche sont entrepris dans un bon nombre d'universités nord-américaines comme l'Université de la Colombie-Britannique, la Penn State University et l'Austin University pour n'en nommer que quelques-unes.

### 1.3 Le projet Hélios

En 1998, quatre anciens membres du projet de sous-marin à propulsion humaine OMER ont fondé l'équipe Hélios HPH à l'École de Technologie Supérieure. L'objectif

premier de l'équipe est bien entendu de tenter de relever le prestigieux défi Igor Sikorsky. Composée de quatre étudiants à la maîtrise et d'environ une dizaine d'étudiants au baccalauréat, autant en génie mécanique qu'en génie électrique, l'équipe travaille depuis presque trois ans à la conception et la réalisation d'Hélios. Le projet est totalement financé par les nombreux commanditaires recrutés par les membres de l'équipe. En retour, l'équipe Hélios s'engage à les représenter adéquatement lors d'événements médiatiques.

### 1.3.1 Choix de la configuration

Les premiers mois d'existence du projet Hélios furent consacrés à différentes tâches ayant comme objectif de déterminer la configuration optimale pour un hélicoptère à propulsion humaine. Les principaux aspects étudiés sont le nombre de pédaleur idéal pour assurer la propulsion de l'engin et la disposition de l'assemblage nacelle-rotors.

#### 1.3.1.1 Nombre de pédaleur

La première étape de design du projet est le choix d'une configuration optimale d'HPH. Puisqu'il est clairement indiqué au règlement 4.2.1 que le nombre de personnes à bord de l'hélicoptère est illimité, une étude portant sur le rendement d'un HPH à plusieurs occupants était de mise. Un banc d'essais pédaleur permettant de mesurer avec exactitude la puissance développée par un équipage de un à sept pédaleurs nous a permis de recueillir des données pour déterminer le choix optimal. Comme on peut le voir à la figure 1.4, les sept pédaliers sont reliés ensemble à un moteur électrique permettant de créer une résistance mécanique variable pour simuler l'effort que devra développer le ou les pédaleurs à bord de l'hélicoptère. L'acquisition de la puissance générée se fait à

l'aide d'un ordinateur par l'intermédiaire d'un couple-mètre et d'un capteur de vitesse de rotation. Avec la relation suivante, il est possible de calculer la puissance.

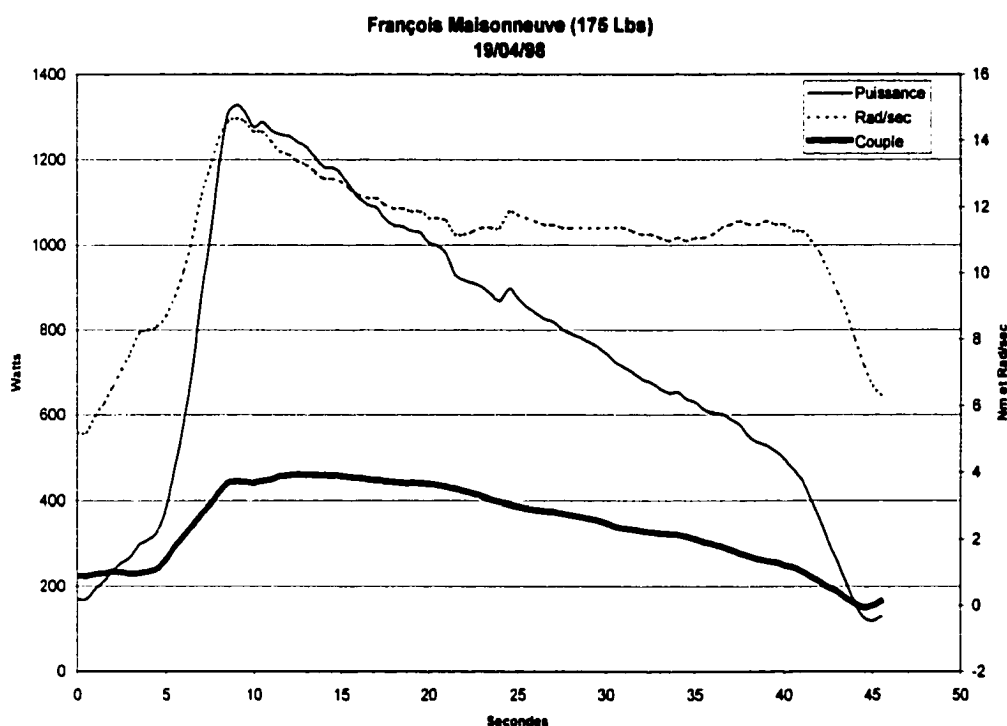
$$P = \omega \cdot T \quad (1.1)$$

La variable  $P$  représente la puissance en Watts,  $\omega$  est la vitesse de rotation en radian par seconde et  $T$  est le couple en Newton-mètre. Le couple-mètre est situé entre le moteur et le boîtier d'engrenage tandis que le capteur de rotation est en fait un capteur de proximité situé sur l'engrenage et générant un train d'impulsions.



Figure 1.4 Banc d'essais pédaleur

Le graphique 1.1 nous montre un exemple des résultats obtenus avec un seul pédaleur lors d'un effort de 60 secondes sur le banc d'essais. On y retrouve le couple généré, la vitesse de rotation et la puissance résultante.



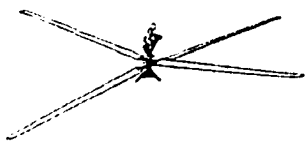
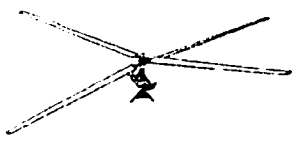


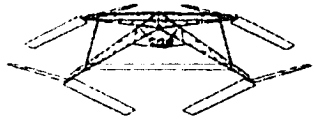
Graphique 1.1 Résultat d'acquisition du banc d'essais pédaleur

Les résultats de puissance développée par un ou plusieurs pédaleurs, recueillis sur le banc d'essais, combinés avec les calculs de poids et de dimension de l'HPH nous ont permis de conclure que l'équipage optimal pour un hélicoptère à propulsion humaine est d'un seul pédaleur.

#### 1.3.1.2 Disposition de la nacelle

Pour choisir une configuration optimale pour notre HPH, nous avons réalisé une revue des différentes dispositions utilisées par les autres équipes jusqu'à ce jour. On retrouve au tableau 1.1 les configurations étudiées ainsi que les avantages et désavantages de chacune.

**Tableau 1.1**  
**Configurations étudiées**

<i><b>Configuration</b></i>	<i><b>Description</b></i>	<i><b>HPH</b></i>	<i><b>AVANTAGES ET DÉSAVANTAGES</b></i>
	<ul style="list-style-type: none"> <li>- Rotors coaxiaux contrarotatifs</li> <li>- Nacelle placée au-dessus des rotors</li> </ul>	<ul style="list-style-type: none"> <li>- Day fly</li> <li>- Papillon A</li> <li>- Vertigo</li> </ul>	<p>A: La configuration maximise l'effet de sol</p> <p>D: L'interférence entre les deux rotors lors des croisées crée des problèmes d'instabilité</p> <p>Si les pales restent horizontales, l'appareil est naturellement instable puisque le centre de gravité de l'appareil est situé au-dessus du centre de lift</p>
	<ul style="list-style-type: none"> <li>- Rotors coaxiaux contrarotatifs</li> <li>- Nacelle placée en dessous des rotors</li> </ul>	<ul style="list-style-type: none"> <li>- Papillon B</li> </ul>	<p>A: La configuration maximise la stabilité</p> <p>D: L'interférence entre les deux rotors lors des croisées crée des problèmes d'instabilité</p>
	<ul style="list-style-type: none"> <li>- Rotor principal avec un rotor de queue.</li> </ul>	<ul style="list-style-type: none"> <li>- Aucun</li> </ul>	<p>A: Principe éprouvé et simple</p> <p>D: 10 à 15% d'énergie perdue dans le rotor de queue</p>
	<ul style="list-style-type: none"> <li>- Rotor à deux pales mis en rotation par deux hélices de bout de pale</li> </ul>	<ul style="list-style-type: none"> <li>- DaVinci III</li> <li>- Yuri II</li> </ul>	<p>A: Aucun couple n'est transmis directement à l'arbre du rotor, donc le contrôle en rotation de la nacelle n'est pas nécessaire</p> <p>D: Rendement faible puisque la puissance n'est pas transmise directement au rotor</p>
	<ul style="list-style-type: none"> <li>- Deux rotors tournent en sens horaire</li> <li>- Deux rotors tournent en sens anti-horaire</li> </ul>	<ul style="list-style-type: none"> <li>- Yuri I</li> </ul>	<p>A: Bonne exploitation de l'effet de sol</p> <p>Configuration naturellement stable</p> <p>D: Système de propulsion peu efficace</p> <p>Beaucoup de composantes ne contribuant pas à la poussée</p>

Il est intéressant de souligner qu'aucun HPH développé depuis vingt ans n'utilise un système de contrôle et de positionnement. Les deux seules équipes à avoir prit leur

envol, ont mis l'emphasis sur la conception ultra légère et sur l'optimisation de la puissance fournit par le pédaleur, mais elles se sont toutes deux butées au problème de stabilité en vol ce qui leur a empêché de respecter le règlement 4.4.1 qui stipule que l'hélicoptère doit rester à l'intérieur d'une surface de 100 mètres carré.

Suite à cette revue de la littérature, le choix de la configuration pour l'hélicoptère Hélios a été déterminé en faisant des compromis entre les avantages et les désavantages des dispositions des autres équipes. Ainsi, une structure composée d'une nacelle au centre de deux rotors coaxiaux contre rotatif a été choisie. Comme on peut le voir à la figure 1.5, le rotor du bas, situé très près du sol, maximise l'effet de sol donnant ainsi une poussée supplémentaire à l'aéronef tout au long du vol. Par ailleurs, le rotor supérieur est situé 1.6 mètres plus haut, réduisant ainsi les dangers d'interférence entre les deux rotors lors du croisement des pales de 17,5 mètres chacune [5]. Le pilote à bord de l'hélicoptère devra pédaler à environ 70 rotations par minutes (RPM) entraînant ainsi les rotors à 7 RPM pour générer la poussée de requise.

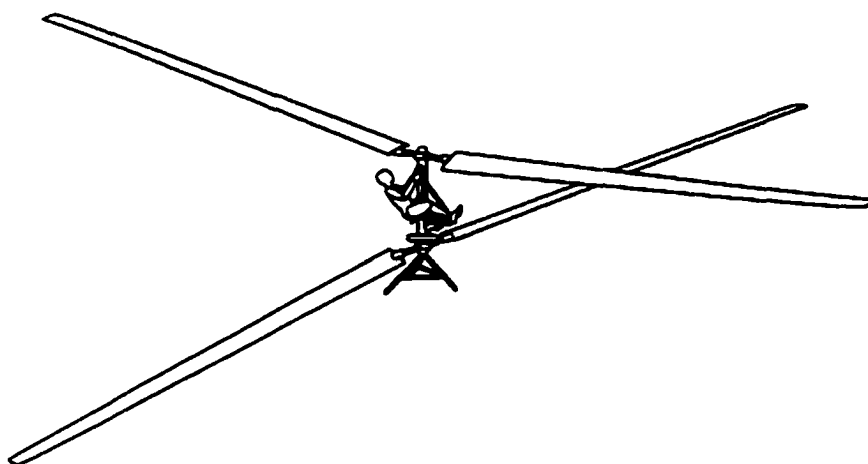


Figure 1.5 Configuration de l'hélicoptère à propulsion humaine Hélios



### 1.3.2 Théorie de poussée des pales

Une bonne partie du succès d'un HPH réside sans aucun doute dans le système de propulsion. À cet effet l'équipe Hélios s'est penchée sur le sujet pour concevoir un système efficace et fonctionnel. Avec une bonne compréhension de la théorie de poussée des pales et une approche innovatrice, nous avons développé un système de propulsion combinant la force mécanique à la précision d'un contrôle électronique.

#### 1.3.2.1 Structure de la pale

Un des premiers critères de stabilité de notre HPH est que le vecteur de poussée résultant de chacun des rotors soit vertical, pour éviter tout déplacement. Cette poussée est générée par les pales en rotation dans l'air. Sans entrer en détail dans la théorie du calcul de poussée d'une pale [5], il est important de noter que l'angle d'attaque de la pale et la vitesse de celle-ci dans l'air sont des facteurs influençant la poussée obtenue. Comme la vitesse en bout de pale est beaucoup plus élevée que la vitesse à l'implanture, la structure de la pale comporte un vrillage donnant un angle de calage supérieur à l'implanture. Ceci permet de répartir la poussée le long de la pale.

#### 1.3.2.2 Système à pas variable

Comme il a été mentionné précédemment, les pales ont une longueur de 60 pieds. Elles sont principalement composées de polystyrène expansé et de fibre de carbone pour un poids total d'environ 35 kilogrammes. Le long et complexe processus de fabrication des pales [5] implique qu'il est pratiquement impossible de fabriquer deux pales

complètement identiques. Par conséquent, lorsqu'elles seront mises en rotation à une vitesse constante avec le même angle d'attaque, elles ne généreront pas la même force de poussée verticale. Ceci a pour effet de compromettre la stabilité de l'hélicoptère si nous adoptons une mécanique à angle d'attaque fixe pour les pales.

Hélios est par conséquent équipé d'un système unique en son genre pour un HPH : un système à pas variable pour modifier l'angle d'attaque des pales. En plus de palier aux imperfections de fabrication, ce système maximise la puissance fournie par le pédaleur aux rotors et ce tout au long du vol. En effet, lors du décollage, le pilote doit développer une certaine puissance pour accélérer les rotors à la vitesse de design de 7 tours par minutes. Durant cette période, aucune poussée n'est requise par les pales. Par conséquent, l'angle d'attaque de chacune des pales est réduit pour rester le plus efficace possible en minimisant l'énergie nécessaire par le pédaleur. Une fois la vitesse de rotation de pointe atteinte, l'angle d'attaque sera augmenté pour faire décoller l'aéronef.

### 1.3.3 Système de positionnement

Contrairement à un hélicoptère commercial, un HPH ne doit pas se déplacer. En effet selon les règlements du concours Igor Sikorsky, le vol d'une minute doit être stationnaire. C'est pourquoi tout le design d'Hélios est axé sur la stabilité de l'engin. Toutefois, la fabrication artisanale des différents composants de l'hélicoptère fait en sorte que plusieurs imperfections peuvent se glisser durant la confection et éloigner ainsi le produit final du design théorique. Par conséquent, la modélisation d'un HPH est très difficile à réaliser ou incomplète. Ceci nous oblige donc à développer une méthode efficace pour déterminer la position de l'appareil en tout temps. De cette façon il sera possible de corriger la position, évitant ainsi une disqualification assurée; un sort qu'a subit plusieurs équipes au cours de ces dernières années.

#### 1.4 Structure du mémoire

Le présente mémoire se veut une synthèse de la conception de l'unité de contrôle et du système de positionnement. Le sujet est divisé en trois chapitres bien distincts en plus de ce chapitre de mise en situation. On discutera en premier lieu du module de contrôle, centre nerveux de l'hélicoptère. C'est en effet cette unité qui assure le contrôle et le monitoring en temps réel lors du vol de l'HPH. Par la suite une présentation du banc d'essais rotor nous permettra de comprendre la philosophie de contrôle avant-gardiste adoptée par l'équipe. En troisième point, le système de positionnement est expliqué. On y discute principalement des différentes approches étudiées, des contraintes et des choix qui ont été faits ainsi que de la stratégie retenue. Finalement, une analyse des résultats ainsi qu'une liste de recommandations conclut ce mémoire.

## CHAPITRE 2

### MODULE DE CONTRÔLE

La revue de la littérature des hélicoptères à propulsion humaine nous a permis d'identifier la problématique dans la conception de ce type d'engin: maximiser la poussée verticale, pour arracher l'appareil à la gravité, en bénéficiant seulement de la puissance réduite d'un athlète humain. Pour parvenir à cette fin, l'équipe Hélios innove dans le domaine en misant sur une stratégie avant-gardiste utilisant un module de contrôle électronique pour assurer un transfert de puissance optimal entre les jambes du pédaleur et les deux rotors de l'hélicoptère. De plus, cette approche permet d'intégrer à l'aéronef différents capteurs et actionneurs pour assurer la stabilité et le positionnement de celui-ci. Le présent chapitre décrit en détail le module de contrôle ainsi que les différents capteurs et interfaces développés autour de celui-ci. Dans un premier temps, la méthodologie de contrôle adoptée pour le projet Hélios sera présentée. Suivra une description détaillée de l'unité de contrôle même avec ses actionneurs et ses différents capteurs. En conclusion de ce chapitre, l'algorithme de vol choisi est exposé ainsi qu'une courte discussion sur l'implantation de celui-ci.

#### 2.1 Méthodologie de contrôle

L'élaboration d'une méthode de contrôle pour un hélicoptère à propulsion humaine est relativement différente de celle utilisée pour les hélicoptères commerciaux.

Premièrement notre HPH est conçu pour maintenir une position stationnaire. Ainsi le système de contrôle doit être optimisé pour assurer la plus grande stabilité à l'engin pour qu'il conserve sa position de départ. De plus, la puissance disponible pour propulser l'appareil étant réduite et momentanée, une attention particulière sera de mise pour maximiser le transfert d'énergie du pédaleur aux deux rotors. Ces derniers étant identiques et indépendants, la suite de cette analyse sera axée sur le développement d'une méthode de contrôle pour un seul rotor composé de deux pales. Par la suite, le système conçu sera utilisé pour chacun des rotors.

### 2.1.1 Théorie de poussé des pales

La poussée verticale générée par les deux pales est fonction de plusieurs paramètres [5]. Deux d'entre eux retiennent notre attention du point de vue contrôle: la vitesse de déplacement de la pale et l'angle d'attaque du profil dans l'air. En variant la vitesse de rotation du rotor, la grandeur du vecteur de poussée verticale est directement influencée. Tout comme sur un hélicoptère commercial, on peut effectivement accroître la poussée verticale en faisant tourner le rotor plus rapidement. Toutefois, cette technique nécessite une puissance supplémentaire du moteur principal, non-disponible avec un HPH. Ainsi, il est essentiel de développer un design et une méthodologie de contrôle composant avec cette contrainte de puissance. Comme il a été mentionné au chapitre précédent, les pales d'Hélios ont une longueur de 17,5 mètres pour un diamètre de rotor de 35 mètres. Cette grande surface balayée par le rotor permet une vitesse de rotation relativement faible de 7 RPM. Celle-ci est constante tout le long du vol et demande au pédaleur, par le biais d'un ratio rotor-pédalier d'environ 1:14.3, de maintenir une cadence de 100 RPM. C'est à ce rythme que le pédaleur a produit une puissance maximum et un effort optimal sur des essais de 60 secondes réalisés sur le banc d'essais pour pédaleurs.

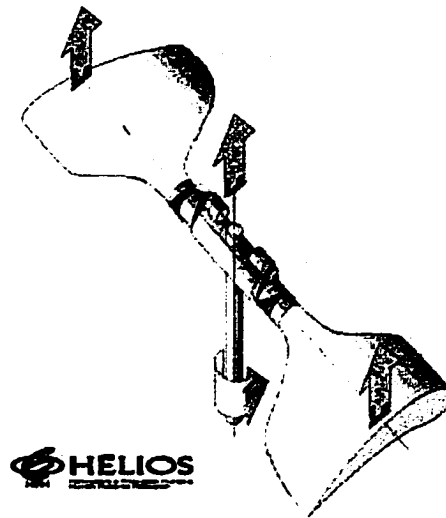


Figure 2.1 Poussées générées par les pales

On peut voir à la figure 2.1 ci-dessus un dessin illustrant le rotor mis en rotation par un couple  $T_R$ . Chaque pale avance dans l'air à une certaine vitesse générant ainsi deux vecteurs de poussée  $L_B$ . Le vecteur de poussée résultant  $L_R$  est situé au centre du rotor et se doit d'être parfaitement vertical pour assurer une bonne stabilité.

Par conséquent, le contrôle de la poussée pour soulever l'engin sera assuré par la variation de l'angle d'attaque des pâles. Cette technique, déjà mise à l'épreuve dans le contrôle du pas variable du sous-marin OMER 3 [8], permet de contrôler efficacement la demande en puissance de chacun des rotors du décollage à l'atterrissage et ouvre la porte sur la possibilité d'assurer une stabilité continue de l'appareil en vol. En effet, en contrôlant l'angle d'attaque " $\alpha$ " de chaque pale de façon indépendante, il est possible d'ajuster la poussée  $L_B$  de chacune des pales pour obtenir un vecteur de poussée  $L_R$  le plus vertical possible.

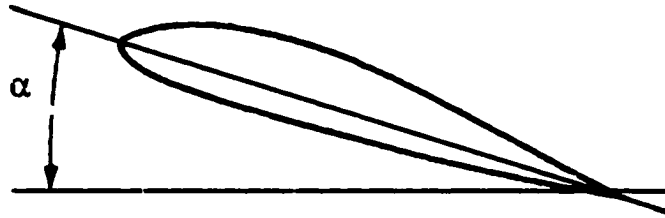


Figure 2.2 Angle d'attaque de la pale

D'autre part, la méthode de fabrication manuelle des pales [5] rend presque impossible l'obtention de pales identiques. En effet, plusieurs étapes sont nécessaires pour la confection de ces dernières, dont des cuissons dépassant les huit heures, laissant place à l'erreur et aux imperfections. Ainsi, pour une vitesse de rotation fixe du rotor et un angle d'attaque identique de chaque pale, la poussée de ces dernières risque fortement d'être disparate, rendant ainsi l'appareil instable. C'est ainsi que le contrôle indépendant de l'angle d'attaque des pales devient indispensable pour permettre de balancer la poussée générée.

### 2.1.2 Boucle de contrôle

L'objectif du module de contrôle est de maintenir le vecteur de poussée résultant du rotor le plus vertical possible de façon temps réel. L'élément à contrôler est l'angle d'attaque, ou le pas, de chaque pale par le biais de servomoteurs. Deux alternatives de contrôle s'offrent à nous: un contrôle en boucle fermée ou un contrôle en boucle ouverte. Avant de choisir l'une ou l'autre de ces méthodes, il est impératif de revoir les différents besoins à combler pour l'élaboration de chacune des boucles de contrôle.

### 2.1.2.1 Boucle fermée

Le contrôle de l'orientation du vecteur poussée  $L_R$ , en utilisant une méthode à rétroaction, nécessite une lecture de celui-ci en temps réel pour calculer l'erreur relative et apporter une correction. Une façon de mesurer le module et l'orientation de ce vecteur est d'utiliser trois cellules de charge disposées de façon triangulaire. La mesure de force de chaque cellule nous permet de calculer par triangulation les composantes caractérisant le vecteur de poussée qui en résulte. Une description détaillée de cette technique fait l'objet d'une section du chapitre suivant. La boucle de contrôle est par conséquent composée de cellules de charge en entrée et d'une consigne de vecteur  $L_R$  vertical. La sortie de cette boucle est une commande pour chaque moteur ajustant le pas des pales.

Par ailleurs, pour balancer la poussée des pales, deux alternatives se présentent: augmenter le pas de la pale qui pousse le moins ou diminuer le pas de la pale qui pousse le plus. En augmentant le pas d'une pale, on augmente directement la demande en puissance du pédaleur. Ceci a pour effet d'être plus exigeant pour ce dernier et une baisse de la cadence peut s'en résulter. Une diminution du pas d'une pale produit le phénomène inverse. C'est pourquoi une lecture du nombre de rotation du pédalier par minute "RPM" est nécessaire pour s'assurer que le pédaleur maintient une cadence cible de 100 RPM.

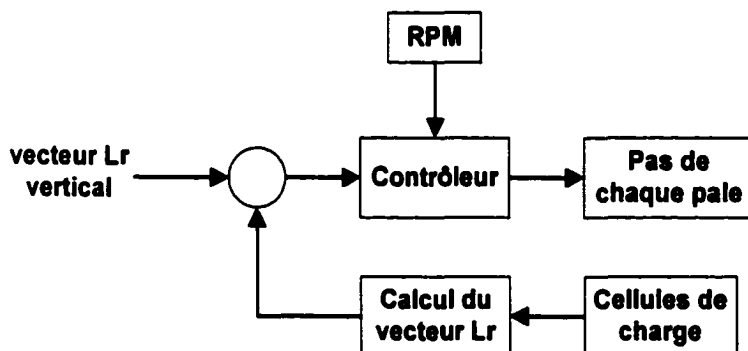


Figure 2.3 Boucle de contrôle fermée



Bien que le contrôle en boucle fermée assure une précision sur la poussée générée par les rotors et une exactitude théorique du contrôle, il est important de remarquer la lenteur du processus. En effet, comme les pales font une rotation complète toutes les 8,5 secondes, il va sans dire qu'une modification du pas d'une pale aura un effet mesurable que quelques secondes plus tard. Ceci a pour effet de rendre le système difficile à contrôler engendrant une grande instabilité. De plus, l'ajout de six cellules de charge totalisant 2.72 kg pour mesurer la poussée de chaque rotor, alourdi sérieusement l'appareil compromettant ainsi l'atteinte de l'altitude de trois mètres. C'est pourquoi cette méthode de contrôle sera mise de côté au profit d'une alternative proposant un temps de réponse beaucoup plus rapide et un poids ajouté à l'aéronef presque négligeable.

#### 2.1.2.2 Boucle ouverte

Le contrôle en boucle ouverte implique des ajustements à l'aveuglette. En effet aucune mesure de la poussée n'est effectuée, empêchant ainsi de valider les corrections apportées. Cette méthode de contrôle est efficace seulement si une modélisation précise du système est possible ou si un apprentissage adéquat des conditions du système a été effectué préalablement. Dans le cas d'un HPH, il est très difficile ou pratiquement impossible de modéliser le système complet, car celui-ci est fortement non-linéaire. Par conséquent, une approche plus pratique sera mise de l'avant, basée sur l'utilisation d'un simulateur pour recueillir les paramètres caractérisant le vol complet de l'hélicoptère. En effet, puisque l'HPH réalise un vol dans un environnement contrôlé sur une période de temps fixe et qu'il exécute des déplacements connus à l'avance, il est possible de recréer ces conditions sous forme d'un simulateur ou d'un banc d'essais. Ce simulateur nous permet alors d'obtenir le pas de chaque pale pour balancer la poussée, en fonction de deux paramètres d'entrée, soit le RPM et la hauteur de l'appareil. Ces données sont ensuite comptabilisées sous forme de tableau.

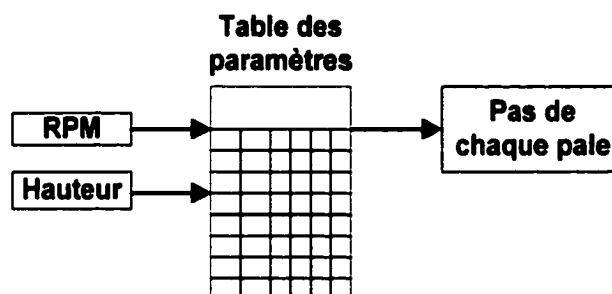


Figure 2.4 Boucle de contrôle ouverte

Cette méthode de contrôle est rapide puisque les valeurs des angles d'attaque sont préalablement calculées. De plus, comme aucune boucle de rétroaction est nécessaire, on élimine le besoin des cellules de charge, conservant ainsi la légèreté de l'appareil. Cette façon de contrôler l'HPH est donc adoptée et sera intégrée dans un module de contrôle responsable de faire l'acquisition des capteurs et de commander les moteurs servant à ajuster le pas de chaque pale.

## 2.2 Unité de contrôle

La méthodologie de contrôle étant exposée, la discussion portera maintenant sur l'unité de contrôle, cœur même du module de contrôle. Il faut en effet concevoir une unité capable de faire l'acquisition des différents capteurs, la gestion de l'algorithme de vol et de commander les moteurs. Comme des données expérimentales seront recueillies sur le simulateur et utilisées par le programme de contrôle, il sera nécessaire de développer une unité avec mémoire pour stocker ces informations. C'est pourquoi un ordinateur de bord à base d'un microcontrôleur fera office d'unité de contrôle.

### 2.2.1 Choix et description de l'ordinateur de bord

Le projet Hélios ayant un échéancier de deux ans et puisque la conception du module de contrôle débuta après la première année d'existence du projet, le choix de l'unité de contrôle s'est porté sur l'ordinateur de bord qui équipe le sous-marin Omer 3. De plus, cet ordinateur rallie performance, modularité, flexibilité et fiabilité [8]. L'aspect modulaire et flexible de cet ordinateur permet de concevoir des cartes d'extension propres à notre application. De plus, ayant été mis à l'épreuve plusieurs fois lors de compétitions, la fiabilité de ce dernier sera une économie de temps de développement non négligeable pour notre projet.

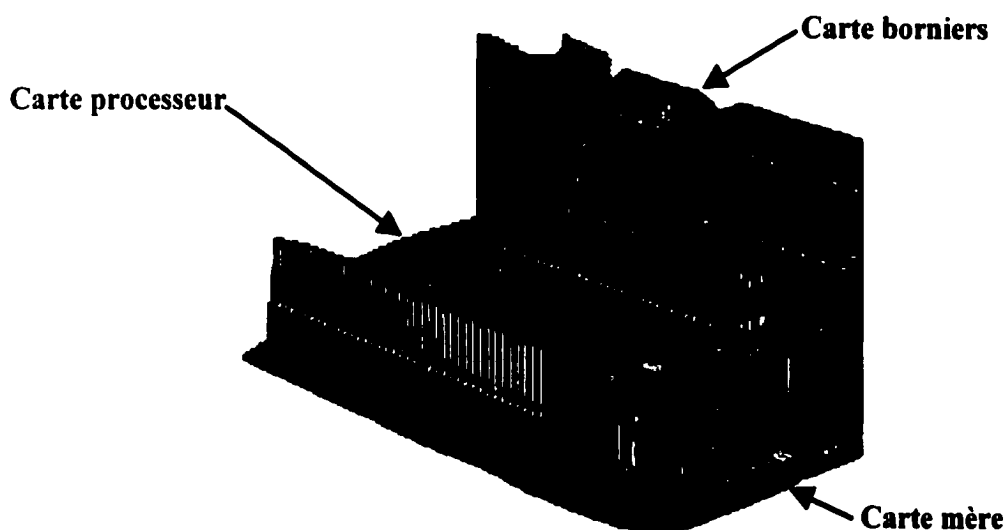


Figure 2.5 Ordinateur de bord Hélios-CPU

L'ordinateur de bord de Hélios est baptisé Hélios-CPU. Celui-ci est composé de trois cartes principales: la carte mère, la carte borniers et la carte processeur. D'autres cartes intégrant les interfaces pour les différents capteurs seront ajoutées par la suite. La carte processeur est composée d'un microcontrôleur MC68F333 de Motorola cadencé à

16 Mhz et d'une mémoire principale de 128 Ko. Deux connecteurs de 72 broches relient cette dernière à la carte mère responsable de l'alimentation et des interconnexions avec les autres cartes d'extension. La carte borniers quant à elle offre 32 points de branchements pour connecter les différents capteurs et actionneurs. Cette unité de contrôle repose entièrement sur une technologie de montage en surface, optimisant ainsi son poids pour une plus grande légèreté.

D'autre part, du point de vue logiciel, cet ordinateur autonome est muni d'un outil de déverminage intégré en microcode permettant une programmation efficace et rapide. Un logiciel sur Pc accompagne ce dernier pour suivre en temps réel le programme et permettre la détection et la localisation des erreurs. Le langage C est utilisé pour l'implantation de l'algorithme de contrôle et pour la gestion temps réel du système.

### 2.2.2 Disposition de l'unité de contrôle

L'unité de contrôle est principalement composée de l'ordinateur Hélios-CPU et de modules d'entrées-sorties. Ces derniers sont responsables de l'interfaçage entre les capteurs et les actionneurs et l'unité Hélios-CPU. La disposition de ces différents modules nécessite une attention particulière puisque les moteurs servant à l'ajustement du pas de chaque pale sont nécessairement fixés à même les rotors. Ces derniers sont en rotation par rapport à la nacelle où réside l'Hélios-CPU. On se rappelle que le mouvement contre rotatif des rotors empêche la nacelle de tourner sur elle-même.

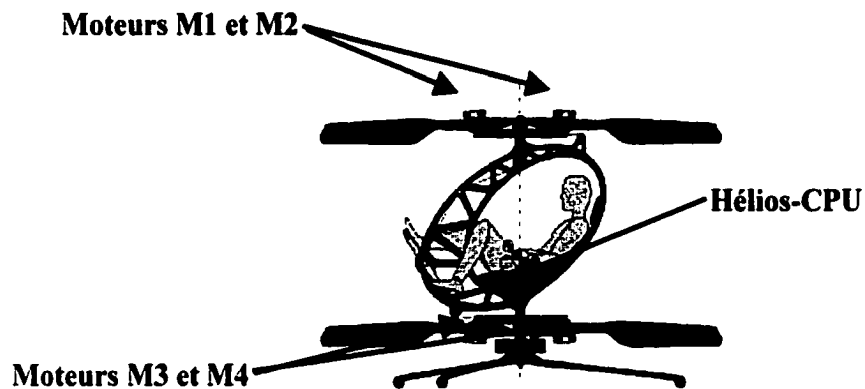


Figure 2.6 Position des moteurs et de l'ordinateur Hélios-CPU

Par ailleurs, le contrôle des servomoteurs, comme il sera discuté dans la section suivante, sera effectué par un contrôle en position. Ainsi, pour minimiser le nombre de signaux à relier du rotor à la nacelle, les cartes de contrôle pour les moteurs seront intégrées aux rotors de façon à réduire le nombre de connexions à quatre par rotor: deux signaux d'alimentation et un signal de consigne pour chaque moteur.

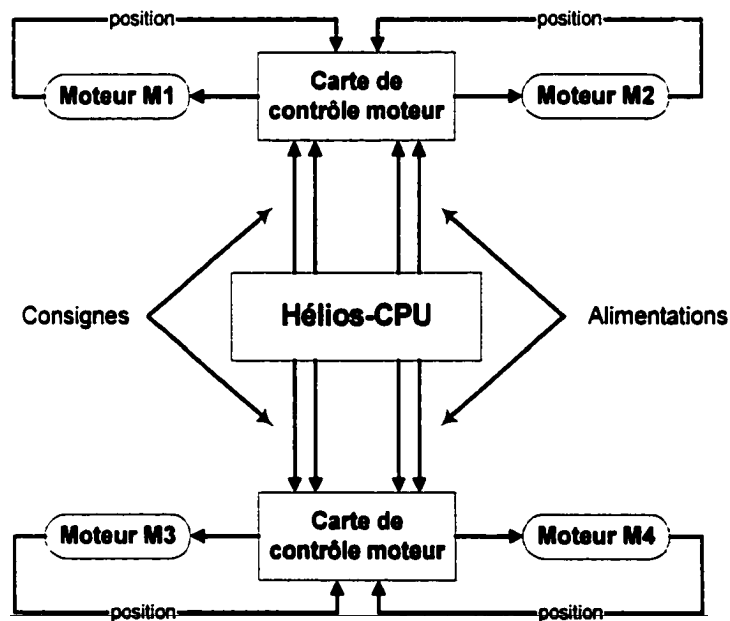


Figure 2.7 Disposition des modules de commande pour les servomoteurs

Il sera donc nécessaire de concevoir deux points de jonction électromécanique pour assurer le passage des signaux électriques de la nacelle aux rotors. Un couplage électrique rotatif [7] a été développé à cet effet par le groupe mécanique.

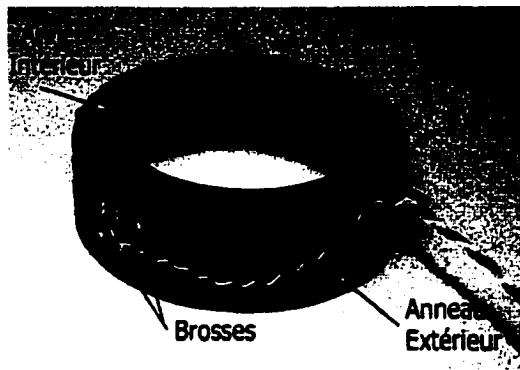


Figure 2.8 Couplage électrique rotatif

### 2.2.3 Entrées-sorties du système

Six modules d'entrée-sortie composent l'unité de contrôle: carte de commande pour les servomoteurs, carte sonar pour mesurer l'altitude, un encodeur pour la mesure du RPM, une carte d'interface RF pour assurer la télémessure en vol, un niveau mètre électronique pour mesurer l'inclinaison de l'appareil et un gyroscope pour évaluer la rotation de la nacelle. Chacun de ces modules est intégré sur des cartes d'extension à même l'ordinateur Hélios-CPU, à l'exception des cartes de contrôle des servomoteurs comme expliqué ci-dessus.

### 2.2.3.1 Carte de contrôle moteur

Lors de la conception du pas variable par l'équipe mécanique [7], un moteur de la compagnie Maxon a été sélectionné pour assurer la rotation des pales pour en changer l'angle d'attaque. Après plusieurs recherches, cette compagnie s'est avérée avoir le meilleur produit combinant puissance et légèreté. Le moteur choisi, de 16 mm de diamètre, tourne à 15 000 tours par minute et offre un couple maximum de sortie de 0,3 Nm par l'intermédiaire d'un réducteur de vitesse 84:1. La consommation moyenne de puissance de ce dernier est de 4,5 Watts. Le moteur est assemblé à une visse sans fin qui entraîne une roue d'engrenage pour provoquer la rotation de la pale.



Figure 2.9 Moteur Maxon assemblé sur le système à pas variable du rotor.

Le contrôle sera assuré par la carte de contrôle moteur [6]. Cette dernière utilise la lecture d'un potentiomètre, fixé au système de pas variable, pour obtenir la position actuelle de la pale. Cette donnée est ensuite comparée à la consigne provenant de l'ordinateur de bord pour déterminer la commande à envoyer au moteur. Pour être en mesure de fournir une consigne analogique à la carte de contrôle moteur, un convertisseur numérique/analogique (DAC) a été utilisé. Le modèle 8 bits TLV5628 de la compagnie Texas Instrument dispose de huit sorties analogiques 0-5 volts. Quatre de

celles-ci seront utilisées comme consigne aux moteurs Maxon et le reste sera disponible pour des applications d'expansion. Par ailleurs, ce convertisseur utilise une communication numérique série, facilitant son intégration avec l'ordinateur de bord par le lien QSPI du microcontrôleur.

### 2.2.3.2 Altimètre

Un des paramètres clés pour l'ajustement du pas des pales est l'altitude de l'appareil. L'approche conventionnelle utilisée dans le domaine de l'aviation utilise la pression atmosphérique. Les hauteurs relativement élevées atteintes par les aéronefs commerciaux font de cette méthode un choix précis et efficace. Toutefois dans le domaine des HPH, d'autres critères doivent être tenus en compte pour assurer la même efficacité. Dans un premier temps, il est intéressant de remarquer que pour des raisons de puissance et de design, un HPH ne dépassera jamais trois mètres d'altitude. Ainsi il devient pensable d'utiliser une technique de télémétrie pour mesurer la distance qui sépare un observateur d'un point éloigné. À ce sujet, l'appareil photo utilise l'ultra son pour mesurer la distance entre l'objectif et l'objet sélectionné. Cette mesure est ensuite utilisée pour faire la mise au point de la lentille. Après quelques recherches, le module 6500 de la compagnie Polaroid semble parfaitement adapté à nos besoins. En effet, cette unité télémétrique permet de mesure avec une précision de  $\pm 1\%$  une distance de 40 cm à 10,67 mètres. C'est ainsi que l'altitude de l'HPH sera en fait obtenue en mesurant la distance séparant le point inférieur de l'appareil avec le sol. Ce module télémétrique utilise un transducteur électroacoustique pour envoyer un ultrason. Ce dernier voyage à environ 338 mètres par seconde. Ainsi, le module envoie un train de 16 impulsions cadencées à 49,4 kHz et attend son retour par écho. Cette unité est pourvue de signaux de contrôle numérique pour l'interfacer aisément avec l'ordinateur de bord.



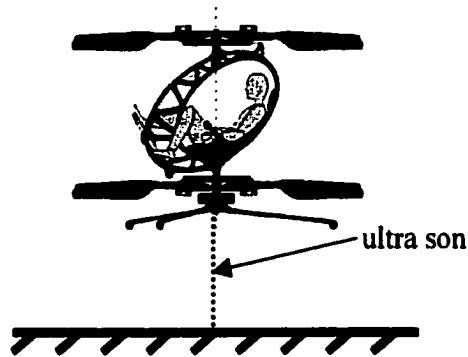


Figure 2.10 Mesure de l'altitude par télémétrie

#### 2.2.3.3 Encodeur

La mesure du RPM s'effectue par l'intermédiaire d'un encodeur directement couplé avec le pédalier. Un encodeur optique de la compagnie Bourns est utilisé. Ce dernier fournit une onde carrée périodique pour mesurer la période de rotation ainsi qu'un signal d'index pour obtenir la position du pédalier et par le fait même des rotors. Ces deux signaux sont de technologie CMOS donc reliés directement avec l'interface TPU du microcontrôleur.

#### 2.2.3.4 Carte de communication RF

Bien que le système de contrôle et de positionnement soit conçu pour être entièrement autonome, une fonctionnalité de télémétrie est implantée pour assurer un

suivi en temps réel lors du vol. En effet, une carte de communication à radiofréquences est intégrée à l'ordinateur de bord pour assurer le transfert des données de l'hélicoptère à la station de contrôle du vol au sol. Cette carte est principalement conçue autour du circuit intégré TR3002 de la compagnie RFM. Ce circuit opère à 418 Mhz et est utilisé pour les communications sans fil à courtes distances. Il offre un taux de transfert de 115.2 kbps sur une distance maximale de 65 mètres, ce qui est amplement suffisant pour notre application. En plus de fournir une fenêtre pour le monitoring du vol, ce lien de communication sans fil permettra au système de positionnement de transmettre les corrections de position qui s'imposent à l'ordinateur de bord.

#### 2.2.3.5 Niveau mètre électronique

L'inclinaison de l'appareil est un paramètre critique en vol. En effet, avec des pales d'une envergure de 17,5 mètres, une faible inclinaison de la nacelle peut provoquer un contact au sol du bout de la pale. Plus la poussée générée par les rotors augmente, plus la déflexion des pales vers le haut est prononcée. Ainsi la nacelle peut se permettre une légère inclinaison en deçà du seuil critique qui varie en fonction de la hauteur et de la poussée de chaque rotor. Pour mesurer cette inclinaison, nous avons recours à une méthode très précise utilisant un accéléromètre pour mesurer une accélération statique, soit la gravité. Le modèle ADXL200 de la compagnie Analog Device permet de mesurer l'accélération selon deux axes perpendiculaires. Conçu sous forme de circuit intégré il s'interface très bien avec le microcontrôleur. Une fois assemblé de façon à ce que ses axes de détection  $A_x$  et  $A_y$  soit parallèle au sol, il ne suffit que d'échantillonner les signaux  $X_{out}$  et  $Y_{out}$  et de mesurer le cycle actif de l'onde carrée. Une fois de plus, les entrées du module TPU du microcontrôleur permettent une mesure précise de cette donnée grâce aux fonctions intégrées. Lorsqu'un axe est complètement perpendiculaire au sol, il subit l'accélération terrestre de  $\pm 1g$  dépendamment de son orientation.

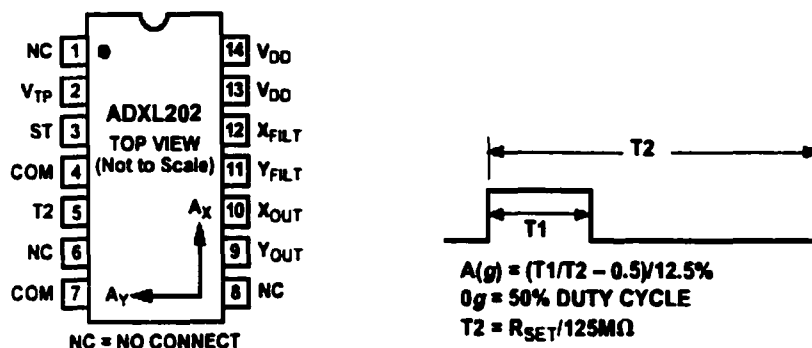


Figure 2.11 Axes de détection de l'accéléromètre et signal de sortie

#### 2.2.3.6 Gyroscope

Le règlement 4.4.3 du concours de l'AHS stipule que la nacelle ne doit pas pivoter sur elle-même de plus d'un tour au cours du vol. C'est pourquoi les deux rotors tournent de façon contre-rotative. Toutefois, encore faut-il que le couple absorbé par chacun des rotors soit le même, sinon une rotation de la nacelle s'en suivra. Si cela se produit, il est possible de corriger la situation en ajustant les angles d'attaque de l'un des rotors, augmentant ou diminuant ainsi la poussée, pour balancer les couples absorbés. La détection de la rotation de la nacelle est possible grâce à l'utilisation d'un gyroscope miniature de la compagnie Gyration. Le Microgyro 100 utilise l'effet de Coriolis pour mesurer la rotation. Avec une résolution de  $0,15^\circ/\text{sec}$ , l'unité fournit un signal analogique proportionnel à la vitesse angulaire. Dans le cas d'un HPH, pour que la nacelle pivote de plus de  $360^\circ$  au cours de la minute de vol, une vitesse angulaire constante d'au moins  $6^\circ/\text{sec}$  devrait être observée. La résolution de gyroscope est par conséquent amplement suffisante. Ce capteur est relié directement au module de conversion analogique-numérique de 10 bits incorporé dans le microcontrôleur. L'échantillonnage constant de ce signal représentant la vitesse angulaire de la nacelle, suivi d'une intégration en fonction du temps de ces valeurs nous donne la position de la nacelle. Cette donnée sera

utilisée par l'algorithme de vol pour apporter un ajustement s'il y a lieu dans le contrôle du pas des pales.

### 2.3 Algorithme de vol

La méthode de contrôle en boucle ouverte utilisée exige un algorithme de vol bien précis. L'ordinateur de bord sera responsable d'ajuster le pas de chacune des pales en temps réel tout au long du vol de 60 secondes. Cette approche nécessite l'utilisation d'une table de données préalablement calculées sur le simulateur. Au moment d'écrire ce mémoire, aucune simulation dynamique sur le banc d'essais n'a encore eu lieu donc les résultats ne sont pas disponibles, mais voici tout de même un exemple de la structure de la table recherchée :

Tableau 2.1  
Exemple de la table de contrôle du pas

Hauteur	RPM	Rotor du bas		Rotor du haut	
		Pale gauche	Pale droite	Pale gauche	Pale droite
$h_0$	$\omega_0$	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$
	$\omega_0 + \Delta$	$\alpha_5$	$\alpha_6$	$\alpha_7$	$\alpha_8$
	$\omega_0 + 2\Delta$	$\alpha_9$	$\alpha_{10}$	$\alpha_{11}$	$\alpha_{12}$
	$\omega_0 + 3\Delta$	...	...	...	...
	$\omega_0 + 4\Delta$	...	...	...	...
$h_0 + \Delta$	$\omega_1$	...	...	...	...
	$\omega_1 + \Delta$	...	...	...	...
	$\omega_1 + 2\Delta$	...	...	...	...
	$\omega_1 + 3\Delta$	...	...	...	...
...	...	...	...	...	...

**Légende**

$h_i$ : hauteur (m)

$\omega_i$ : RPM (tour/sec.)

$\alpha_i$ : angle (°)

$\Delta$ : incrément

L'ordinateur utilise comme index d'entrée dans la table la hauteur de l'hélicoptère et le RPM du pédalier. Étant donné la nature ponctuelle des données de simulation,

l'algorithme devra interpoler pour obtenir des valeurs intermédiaires. Les angles  $\alpha$  sont obtenus de la table, puis le pas des pales du rotor du bas est ajusté en premier suivi de celui des pales du rotor du haut. Suite à ces changements des angles d'attaques des pales, aucune lecture de poussée n'est effectuée pour valider les modifications. L'efficacité de ce contrôle en boucle ouverte repose sur l'exactitude des valeurs comptabilisées dans la table de données.

### 2.3.1 Contrôle de la rotation de la nacelle

Bien que le contrôle de la poussée des pales soit en boucle ouverte et qu'aucune boucle de rétroaction n'est utilisée, le contrôle de la rotation de la nacelle se fait quant à lui à l'aide d'un capteur supplémentaire. Ainsi, la lecture du gyroscope permet d'identifier la grandeur et le sens de la rotation de la nacelle pour fournir à l'algorithme de contrôle l'information supplémentaire permettant de balancer le couple absorbé par chacun des rotors. Si la nacelle subit une rotation de sens horaire, le rotor du bas absorbe donc un couple supérieur au rotor du haut, car le rotor près du sol tourne en sens anti-horaire. Par conséquent, le pas des pales du rotor du haut sera augmenté pour augmenter le couple absorbé par ce dernier, égalisant ainsi les couples absorbés. Dans le cas d'une rotation anti-horaire de la nacelle, le pas des pales du rotor du haut sera diminué. Les ajustements sont exécutés sur le rotor du haut et non sur le rotor du bas, pour conserver la poussée supplémentaire générée par le rotor du bas dû à l'effet de sol [5], minimisant ainsi les pertes de puissances engendrées.

Il faut remarquer ici que le contrôle de la nacelle utilise une boucle de rétroaction pour modifier le pas des pales. Toutefois, le contrôle de la poussée des rotors est initialement exécuté à partir d'une table de donnée et n'utilisant aucune boucle de rétro-

action La figure suivante illustre sous forme d'ordinogramme le processus de contrôle et les différents capteurs impliqués.

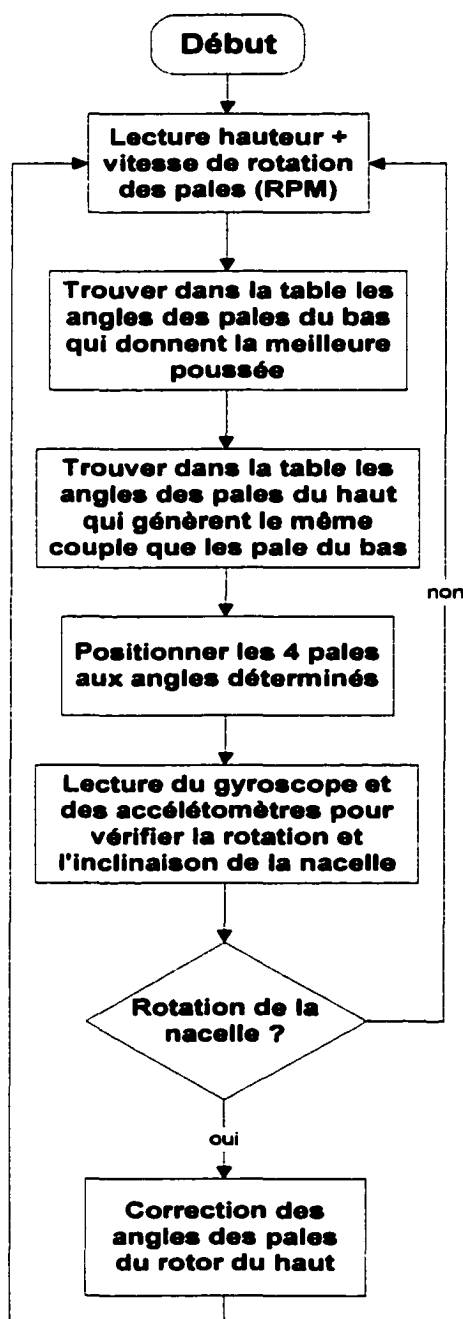


Figure 2.12 Ordinogramme du processus de contrôle

## 2.4 Programmation

L'efficacité et la fiabilité de l'ordinateur de bord ne reposent pas seulement sur sa conception matérielle. Une fois en vol, toute la gestion en temps réel de l'échantillonnage des différents capteurs, du contrôle des actionneurs et de la communication RF demande une programmation rigoureuse et optimisée. C'est pourquoi l'utilisation du langage de programmation C est de mise pour faciliter la segmentation du programme et l'enchaînement en temps réel des routines. La structure de fichier adoptée nous permet de voir les principaux éléments qui composent le programme.

Tableau 2.2  
Segmentation du programme de contrôle en fichiers.

Segmentation en fichier	Description
helioscpu.c	Routines principales: initialisation, main, interruptions, etc.
comm333.c	Routines de gestion de la communication RS-232.
prot333.c	Routines de gestion du protocole de communication.
acqui.c	Routines de gestion de l'acquisition des capteurs.
adc.c	Routines de commande du convertisseur A/D pour les consignes moteurs.
rtc.c	Routines de communication avec l'horloge temps réel.
seprom.c	Routine pour la sauvegarde hors tension des paramètres.

L'exécution du programme se fait en respectant les critères de programmation temps réel et chaque fonction de l'unité de contrôle se voit attribuer une portion du temps d'exécution. Des compteurs de protection (watch dog) sont ajoutés pour éviter tout arrêt du programme dans des routines non vitales au contrôle de l'hélicoptère. À titre

d'exemple, un blocus logiciel au niveau de la communication série serait détecté par un temporisateur et entraînerait tout simplement l'arrêt de la télémesure au profit de la continuité de l'exécution du contrôle du pas des pales. Plusieurs autres routines de protection sont ajoutées au programme pour sécuriser au maximum l'appareil une fois en vol. Des conditions d'atterrissage d'urgence sont éventuellement à déterminer et seront intégrées au programme pour ramener l'hélicoptère au sol de façon sécuritaire.

En conclusion, le module de contrôle est responsable de l'acquisition et du traitement des données lorsque l'hélicoptère est en vol. En suivant une approche de contrôle en boucle ouverte, il échantillonne les différents capteurs puis détermine, à l'aide d'une table de données, l'angle optimal de chacune des pales pour maximiser le transfert en puissance du pédaleur à la propulsion de l'engin. Le succès de celui-ci repose bien entendu sur son design et son implantation, mais surtout sur les données expérimentales recueillies lors des simulations dynamiques sur le banc d'essais. C'est pourquoi la conception électronique du simulateur est un sujet particulièrement important et fait l'objet du prochain chapitre.



## CHAPITRE 3

### SIMULATEUR

Le système de contrôle en boucle ouverte de l'hélicoptère hélios repose essentiellement sur une table de données expérimentales. Des capteurs intégrés à l'appareil permettent de recueillir quelques informations supplémentaires qui servent d'entrée à l'algorithme de contrôle pour choisir les bons angles d'attaque des pales pour que la poussée résultante soit toujours verticale. Ces derniers sont préalablement trouvés à l'aide d'un banc d'essais qui joue le rôle d'un simulateur pour calibrer les pales. Ce simulateur est principalement constitué d'un assemblage mécanique et d'un système d'acquisition électronique permettant de recueillir les données requises pour l'élaboration de la table de contrôle et pour valider le comportement dynamique de l'assemblage pale rotor.

C'est pourquoi ce simulateur joue un rôle capital au sein du projet Hélios et a nécessité plusieurs mois de travail. Le présent chapitre ne se veut pas une description détaillée de tous les aspects du simulateur. L'accent est plutôt mis sur le fonctionnement global du banc d'essais et sur la conception du système d'acquisition. Dans un premier temps, la configuration générale du banc d'essais sera présentée d'un point de vue électromécanique. Le développement du système de contrôle pour assurer la propulsion des rotors fait l'objet du deuxième point, suivi par une description complète du système d'acquisition et de la fonctionnalité de télémesure implantée. En terminant, la procédure de calibrage des pales sera présentée ainsi qu'une courte discussion sur les risques

d'erreur engendrés sur les données lors de ces simulations et de l'impact de celles-ci sur le comportement de l'hélicoptère.

### 3.1 Configuration du banc d'essais

Le banc d'essais rotor, qui fait office de simulateur pour valider et calibrer les pales de l'hélicoptère, est constitué principalement d'une structure métallique sous forme d'élévateur en ciseau. Un arbre est fixé sur cette structure pour supporter deux rotors ainsi que les quatre pales. L'objectif de ce banc d'essais étant d'exécuter une série de tests et de mesures, il n'est pas nécessaire d'avoir recours à un humain pour entraîner les rotors. C'est pourquoi des moteurs hydrauliques sont utilisés pour fournir la puissance nécessaire à la rotation des rotors. De plus cette approche nous permet d'avoir un plein contrôle sur la vitesse de chaque rotor lors des simulations. Par ailleurs, la hauteur ajustable de ce banc d'essais permet, lors des simulations, de balayer toute les hauteurs possibles que peut prendre l'hélicoptère en vol, soit de 0 à 3 mètres. On peut voir à la figure 3.1 le plan technique du banc d'essais.

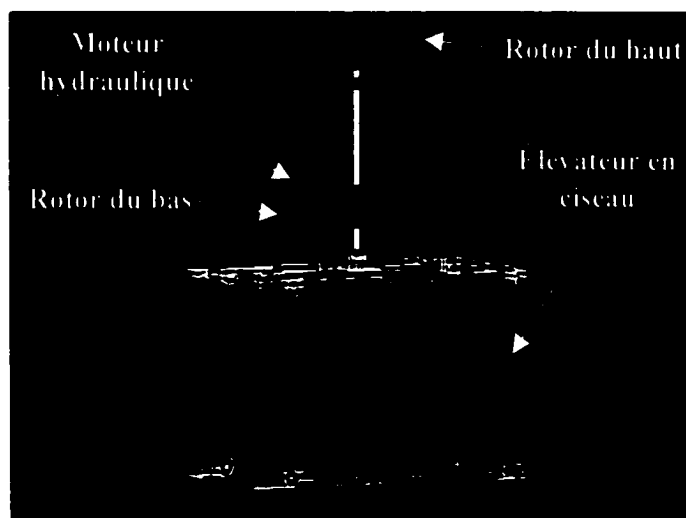


Figure 3.1 Plan mécanique du banc d'essais

Cet assemblage mécanique est bien entendu instrumenté pour assurer la cueillette des données. Étant donné la symétrie qui existe entre les deux rotors, le reste de la discussion portera seulement sur les éléments requis pour l'instrumentation d'un seul rotor. Les mêmes modules et outils développés pourront ensuite être utilisés pour l'instrumentation du deuxième rotor. Le principal objectif étant de calibrer les pales, en ajustant leur angle d'attaque, pour que le vecteur de poussée résultant soit vertical, il devient essentiel de mesurer cette poussée générée.

La méthode utilisée requiert trois cellules de charge disposées de façon triangulaire sur un plateau. Ce dernier est fixé à la structure du banc d'essais et au rotor. Par conséquent, le banc d'essais étant solidement fixé au sol, le rotor mis en rotation exerce des forces d'étirement et de compression sur le plateau de cellule de charge. La cellule de charge est en fait un pont de Wheatstone à quatre résistances. La valeur de ces dernières varie en fonction de la charge exercée changeant ainsi la différence de potentiel présente aux bornes du réseau.

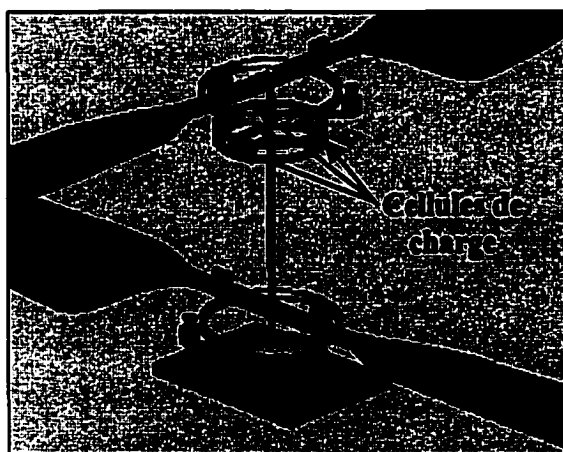


Figure 3.2 Plateaux de cellules de charge

### 3.1.1 Théorie du calcul du vecteur de poussée

La force exercée sur une seule cellule de charge n'a pas vraiment de signification pour notre application. C'est plutôt l'assemblage sous forme de plateau triangulaire de trois cellules qui nous permet de calculer la poussée résultante du rotor. En effet, chaque cellule est positionnée sur un des sommets du triangle à une distance  $r$  du centre de celui-ci. Un angle de  $120^\circ$  est décrit entre chacun des rayons.

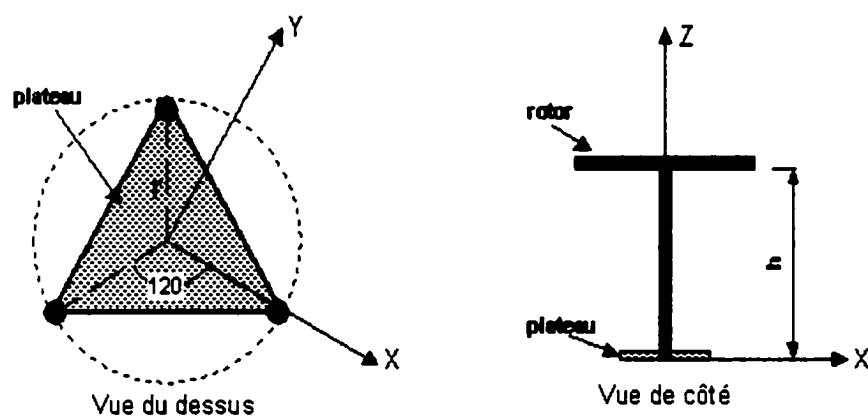


Figure 3.3 Disposition des cellules de charge sur le plateau triangulaire

En sachant que le rotor est positionné de façon perpendiculaire au-dessus de ce plateau à une hauteur  $h$ , il est possible de calculer chacune des composantes du vecteur de poussée selon les axes  $X$ ,  $Y$  et  $Z$  en vérifiant les équations suivantes:

$$\sum F_z = 0$$

$$L_1 + L_2 + L_3 = R_z \quad (3.1)$$

#### Légende

$L$ : force appliquée sur une cellule  
 $R$ : composante du vecteur poussée  
 $r$ : rayon du cercle décrit par les cellules

$$\underline{\sum M_x = 0}$$

$$-R_y h + L_2 r \cos(30^\circ) - L_1 r \cos(30^\circ) = 0 \quad (3.2)$$

$$\underline{\sum M_y = 0}$$

$$R_x h + L_3 r - L_2 r \sin(30^\circ) - L_1 r \sin(30^\circ) = 0 \quad (3.3)$$

La première équation dicte que la somme des forces lues par les cellules de charge est égale à la composante du vecteur de poussée selon l'axe Z. Les deux autres composantes du vecteur de poussée sont obtenues en calculant la somme des moments selon les axes X et Y. Avec ces trois valeurs, le vecteur de poussée est positionné dans l'espace. Il faut remarquer ici que seule la composante en Z de ce vecteur contribue à soulever l'hélicoptère. Il faut donc réduire le plus près de zéro les composantes en X et Y de sorte que le vecteur de poussée soit parfaitement vertical assurant ainsi une stabilité à l'appareil.

### 3.2 Système de contrôle

L'entraînement des rotors étant effectué par des moteurs hydrauliques, nous avons recours à des valves proportionnelles pour contrôler le débit de l'huile et par le fait même la vitesse de rotation du moteur. Un module électronique assurera le contrôle en vitesse de ce moteur par l'intermédiaire de la valve proportionnelle. Ce module, assemblé sous forme de circuit imprimé, reçoit en paramètre d'entrée la vitesse du moteur et la consigne. Il commande alors l'ouverture ou la fermeture de la valve, sous forme d'un signal analogique, jusqu'à ce que la vitesse désirée soit atteinte. Un contrôle proportionnel et intégral est utilisé pour assurer une stabilité acceptable. Voici le schéma de la boucle de contrôle fermée de ce système :

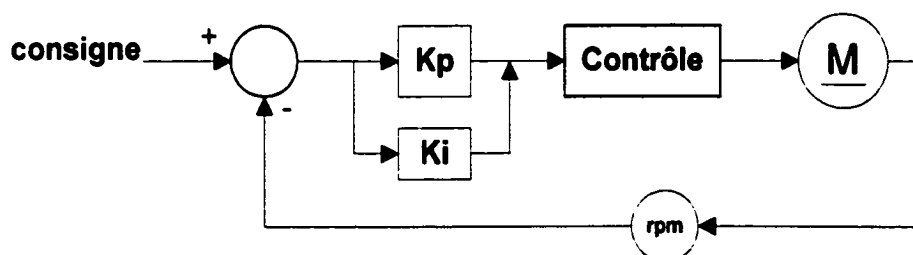


Figure 3.4 Boucle de contrôle fermé du moteur hydraulique

La réalisation matérielle de cette boucle de contrôle a nécessité l'utilisation d'un convertisseur de fréquence à signal continu. Un LM2917 convertit alors le train d'impulsions de l'encodeur en un signal DC. Un amplificateur opérationnel LM324 est utilisé pour exécuter la comparaison entre ce signal DC et la consigne désirée. Le contrôleur proportionnel et intégral,  $K_p$  et  $K_i$ , utilise aussi des LM324 pour fournir en sortie le signal de commande à envoyer à la valve proportionnelle. On retrouve à l'annexe B le plan complet de cette carte d'interface.

### 3.3 Système d'acquisition

Le système d'acquisition du banc d'essais a comme principal mandat de remplir les deux tâches suivantes:

- Échantillonner les différents capteurs.
- Traiter et analyser les données.

Plusieurs approches sont possibles pour réaliser ces fonctions. Des contraintes de coûts, de disposition et d'efficacité ont guidé le choix de la configuration d'acquisition la mieux adaptée à notre application. En ce qui a trait à la disposition, l'envergure des pales

implique que lors d'une simulation, nous devons être situés à au moins 20 mètres du banc d'essais. Par conséquent, pour pouvoir analyser les données en temps réel lors de la simulation, il devient nécessaire de diviser le système d'acquisition en deux parties. La portion acquisition des capteurs est située à même le banc d'essais pour minimiser le filage, tandis qu'une station de travail de type Pc, située à une distance sécuritaire du banc d'essais, est utilisée pour traiter et analyser les données. Un lien de communication RF relie ces deux unités pour assurer le transfert en temps réel des données.

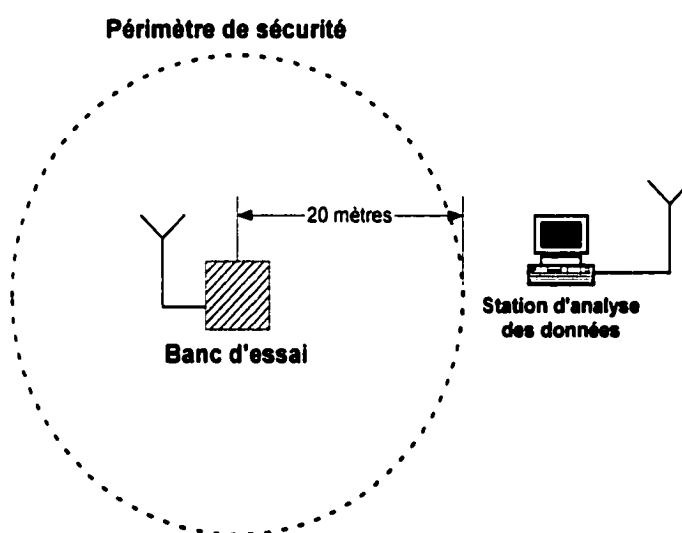


Figure 3.5 Disposition du système d'acquisition

### 3.3.1 Choix de l'unité d'acquisition

Une des fonctions du banc d'essais étant d'ajuster l'angle d'attaque des pales pour produire des simulations, il devient essentiel d'intégrer l'ordinateur de bord de l'hélicoptère au banc d'essais pour s'acquitter de ces fonctions. Cette approche permet de développer les routines de contrôle et les cartes d'interface, nécessaires pour commander les moteurs Maxon, conjointement à la conception du simulateur. De plus, le banc d'essais devient ainsi un outil de test pour valider ces designs.

D'autre part, le système d'acquisition nécessite une unité d'acquisition responsable de l'échantillonnage des capteurs. Pour minimiser les coûts du système et utiliser l'équipement déjà mis en place, l'ordinateur de bord jouera le rôle de cette unité d'acquisition. En effet, le cœur de cet ordinateur étant un microcontrôleur relativement puissant et bénéficiant de plusieurs ports d'entrées-sorties disponibles, il est possible d'interfacer les capteurs au Hélios-CPU et de concevoir un programme temps réel d'acquisition.

De plus, le développement des routines et du protocole de communication pour la télémessure associé au banc d'essais pourra être réutilisé intégralement pour la télémessure de l'hélicoptère même, étant donnée l'utilisation de la même plate-forme matérielle. Finalement, l'ajout d'un module pseudo-intelligent à même le banc d'essais permettra d'y intégrer la génération des consignes pour les cartes de contrôle des moteurs hydrauliques. Il sera alors possible de régler à distance la vitesse des moteurs et de générer des simulations plus complexes exigeant une variation en temps réel de la vitesse des rotors et de l'angle d'attaque des pales.

L'unité d'acquisition occupera donc aussi un rôle de contrôle pour commander les moteurs hydrauliques et les moteurs Maxon des pales. Voici un schéma plus détaillé du système d'acquisition illustrant les différents modules impliqués dans ce design:



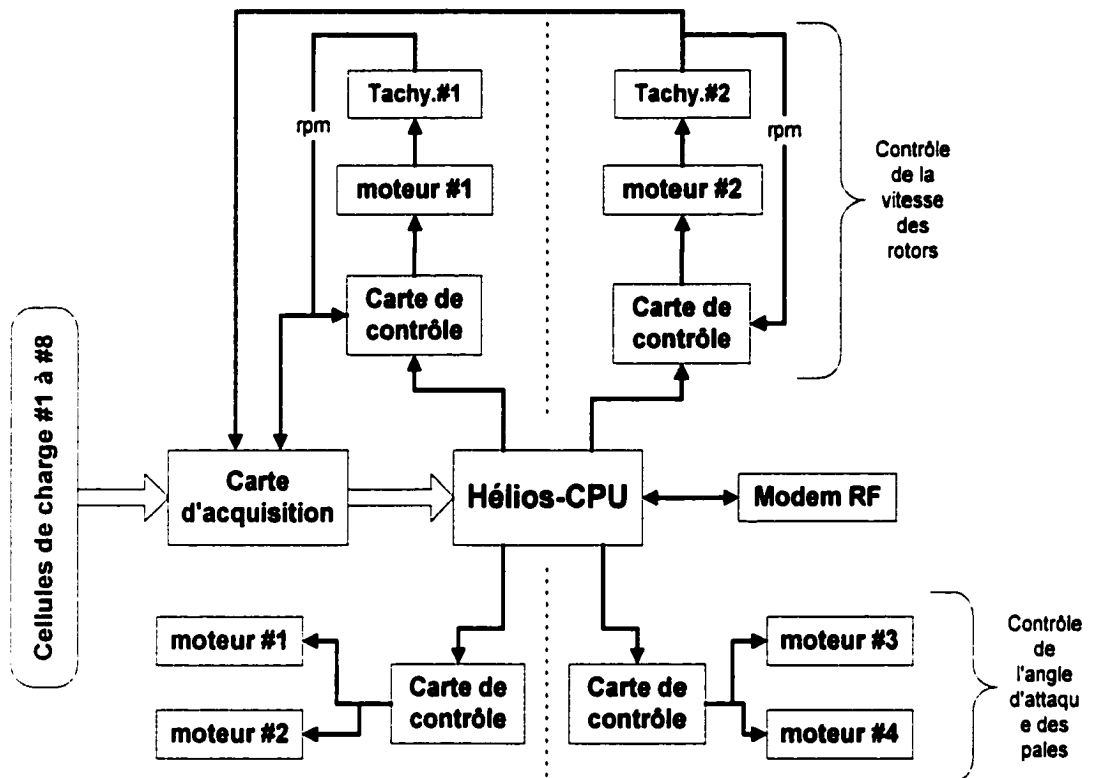


Figure 3.6 Diagramme bloc de l'unité d'acquisition et de contrôle

### 3.3.2 Carte d'acquisition

La connexion des capteurs à l'ordinateur Hélios-CPU exige la conception d'une carte d'interface pour adapter les signaux au microcontrôleur. Principalement les cellules de charge utilisées pour la mesure de la poussée sont munies de signaux de sortie différentiels. L'objectif étant d'échantillonner ces derniers par le biais des convertisseurs analogique-numérique du microcontrôleur opérant sur une plage 0-5 volts, il devient nécessaire de les conditionner. Les cellules de charge utilisées peuvent mesurer une force de  $\pm 90,7$  kg. Une force positive évoque une compression de la cellule et une force négative un étirement. Comme énoncée précédemment, la cellule de charge est en fait un

pont de Wheatstone auquel on applique une tension d'excitation  $V_{in}$  de 12 volts. Parmi les quatre résistances de ce pont, une d'entre elles est en fait une jauge de déformation  $R_g$ . On mesure ainsi entre les bornes de sortie S- et S+ une différence de potentiel  $V_{out}$  proportionnel à la mesure de contrainte de la résistance  $R_g$  et par le fait même de la charge appliquée sur la cellule. Les spécifications de la cellule de charge nous indiquent que la tension de sortie maximale est de 3,224 mV par volts d'alimentation. Ainsi, avec une alimentation de 12 volts, la sortie  $E_o$  sera de  $\pm 38,688$  mV pour une charge de  $\pm 90,7$  kg. Ceci nous donne donc une résolution de  $426,46 \mu V / kg$ .

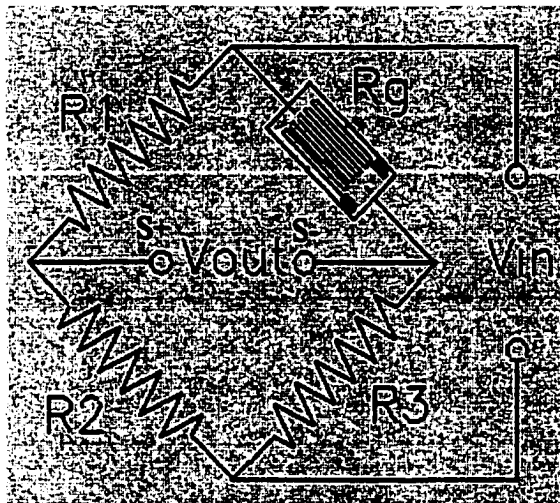


Figure 3.7 Pont de Wheatstone d'une cellule de charge

Le conditionnement du signal de sortie  $V_{out}$  de la cellule de charge a été réalisé à l'aide d'amplificateurs d'instrumentation. Le circuit intégré AD623 de la compagnie Analog Device est parfaitement adapté à notre application. Son principe de fonctionnement est basé sur l'approche classique à trois amplis op, pour convertir un signal d'entrée différentiel de faible amplitude à un signal de sortie continu centré sur une référence configurable.

Pour notre application, les cellules de charge sont en mesure de lire une force en compression ou en étirement. Ceci est représenté par un signal différentiel positif ou négatif en référence aux deux fils de sortie S+ et S- de la cellule. C'est pourquoi la sortie du circuit AD623 est décalée avec une référence de 2,5 volts. De cette façon, l'intervalle 2,5 à 5 volts représente une force de compression et l'intervalle 0 à 2,5 volts représente une force d'étirement sur la cellule de charge. Le logiciel d'analyse des données sera responsable de différencier le type de force selon l'intervalle auquel appartient l'échantillon. Le schéma bloc suivant représente les étapes de conversion du signal différentiel d'une cellule de charge:

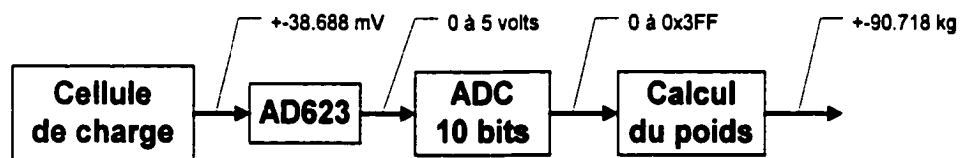


Figure 3.8 Étapes de conversion du signal d'une cellule de charge

Chaque rotor nécessitant trois cellules de charge pour mesurer la poussée générée, la carte d'acquisition sera donc composée de six circuits de conditionnement. De plus, le couple absorbé par chacun des rotors devant être le même pour éviter une rotation de la nacelle, deux autres cellules de charge sont intégrées aux rotors pour mesurer ces forces. Par conséquent, huit signaux sont échantillonnés par le convertisseur analogique-numérique du microcontrôleur.

D'autre part, les mêmes encodeurs qui sont utilisés sur l'hélicoptère pour mesurer la vitesse de rotation des rotors sont intégrés sur le banc d'essais. Ces encodeurs se connectent directement sur les entrées TPU du microcontrôleur.

### 3.3.3 Interface d'analyse de données

L'unité d'acquisition du banc d'essais transmet en temps réel, par le biais du lien de communication RF, les données échantillonnées. Un ordinateur de type Pc reçoit ces dernières et s'occupe du traitement et de l'analyse de celles-ci. Voulant bénéficier de la liberté de programmation qu'offre le langage C et de la rapidité de développement d'une interface graphique qu'offre un outil de conception visuel, le logiciel Lab Windows de la compagnie National Instrument a été choisi pour le développement de l'interface d'analyse de données. Ce logiciel est pourvu d'une vaste librairie de symboles graphiques rattachés à l'instrumentation. Ainsi, le développement de l'aspect visuel du logiciel est facilité et accéléré. On peut voir à la figure ci-dessous un extrait de l'interface d'analyse de données pour visualiser les forces exercées sur les huit cellules de charge.

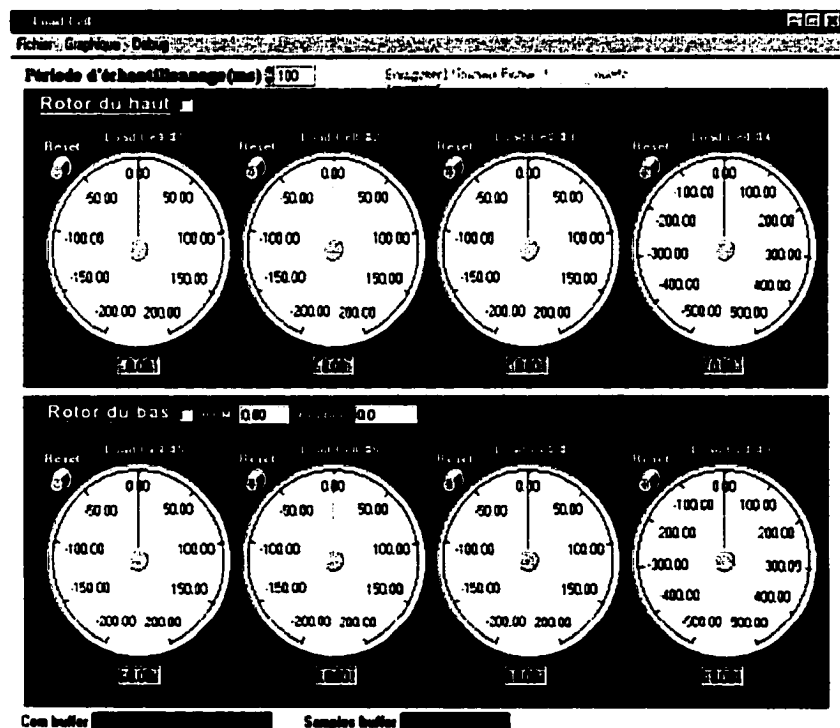


Figure 3.9 Interface Pc pour l'analyse des données

Ce logiciel d'analyse des données offre aussi la possibilité de commander les différents moteurs du banc d'essais. Dans un premier temps, des boutons de contrôle manuel permettent d'ajuster la vitesse de rotation de chaque rotor de façon indépendante. Le logiciel exerce une compensation sur cette commande pour s'assurer de faire accélérer le rotor de façon proportionnelle et sécuritaire. D'autre part, il est aussi possible d'ajuster l'angle d'attaque de chaque pale par l'intermédiaire de ce logiciel. Cette fonctionnalité permettra le calibrage précis de la poussée des rotors.

Finalement, ce programme est pourvu aussi d'un mécanisme de sauvegarde des données. Il est ainsi possible de créer des tableaux de données suite aux simulations et de tracer des graphiques du comportement dynamique de l'assemblage pales rotor. À partir de ces résultats, il sera donc possible de créer la table contenant les valeurs des angles d'attaque, tabulées en fonction du rpm et de la hauteur des rotors, pour maintenir un vecteur de poussée vertical. Cette table sera par la suite intégrée au programme principal de contrôle de vol dans l'ordinateur de bord de l'hélicoptère.

### 3.4 Procédure de calibrage

Lors des simulations, nous voulons ajuster les angles d'attaque des pales pour redresser le vecteur de poussée de façon verticale. Pour ce faire nous avons recours à une procédure de calibrage pour chaque rotor. Voici les principales étapes de cette procédure:

- Fixer l'angle d'attaque de la pale maître.
- Fixer le RPM du rotor à une vitesse constante.
- Varier l'angle d'attaque de la pale esclave jusqu'à ce que le vecteur de poussée soit parfaitement vertical.

- Pour ces valeurs d'angles d'attaque, sauvegarder le rpm, la hauteur, le couple et la poussée.

Cette procédure sera répétée à différents RPM et différentes hauteurs jusqu'à ce que toutes les possibilités soient compilées.

Nous venons de voir que le simulateur, ou banc d'essais de l'assemblage pales/rotors, est un projet en soit qui comporte plusieurs éléments. Les pales sont testées et calibrées sur celui-ci pour recueillir le plus d'information possible. Cet apprentissage nous permet alors d'exercer un contrôle en vol basé sur les données expérimentales de simulation. Bien que le simulateur nous permette d'obtenir une très bonne estimation du comportement des pales en vol, la déformation de celles-ci et les effets difficilement prévisibles de la stabilité de la nacelle font en sorte que l'hélicoptère ne peut être complètement stable et un déplacement de celle-ci sur la surface de vol est prévisible. Par conséquent, il est nécessaire de se doter d'une technique pour déterminer la position exacte de l'appareil et permettre d'apporter les corrections qui s'imposent.

## CHAPITRE 4

### SYSTÈME DE POSITIONNEMENT

Nous venons de voir que le banc d'essais permet de calibrer les pales pour obtenir la poussée optimale tout en conservant la stabilité de l'appareil. Toutefois, des phénomènes dynamiques difficiles à modéliser empêchent l'équilibre absolu de l'aéronef et laissent place au déplacement de celui-ci. Toutefois, le Défi Igor Sikorsky impose avec le règlement 4.4.1 que l'hélicoptère demeure à l'intérieur d'une surface de 100 mètres carrés. Cette contrainte implique qu'il sera impératif de déterminer la position de l'appareil en temps réel lors du vol de 60 secondes pour permettre éventuellement au système de contrôle de repositionner l'HPH. Ce chapitre fait une revue complète du développement du système de positionnement pour l'hélicoptère Hélios. En premier lieu, les principaux paramètres régissant l'aspect du positionnement du vol de l'HPH seront énoncés. Par la suite, une revue de différentes méthodes, pour déterminer la position d'une cible, sera présentée. Et finalement, le reste de ce chapitre sera consacré à une description détaillée du fonctionnement du système de positionnement choisi.

#### 4.1 Paramètres de positionnement de l'HPH

Pour être en mesure de concevoir un système de positionnement efficace, il faut prendre en considération différents paramètres relatifs à l'environnement dans lequel se déroulera le vol de l'HPH. Ces paramètres serviront ensuite de référence pour concevoir le système. Premièrement, l'appareil sera positionné au milieu de la surface de 100

mètres carrés pour le décollage, comme on peut le voir à la figure 4.1. De cette façon, l'hélicoptère possède une liberté de déplacement maximale de cinq mètres suivant l'axe X ou l'axe Y. Dans le pire cas, un déplacement unidirectionnel parallèle à l'arrête du carré peut entraîner l'engin hors surface en seulement 5 mètres. Ce premier paramètre de positionnement est critique seulement si on le conjugue avec la vitesse de déplacement relative de l'appareil et la durée totale du vol, soit 60 secondes.

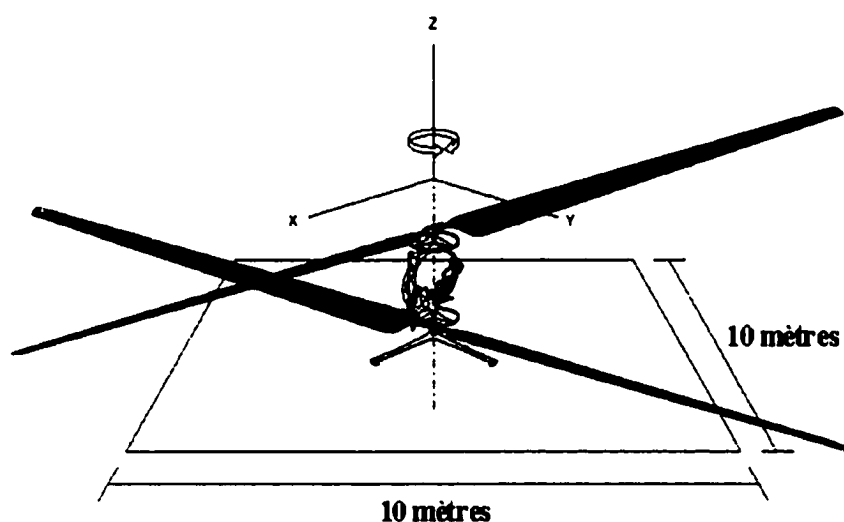


Figure 4.1 Position de décollage de l'HPH

L'hélicoptère étant conçu pour exécuter un vol stationnaire, la vitesse de déplacement est supposée être nulle. Toutefois, l'erreur engendrée lors du calibrage, la déformation des pales en vol ou un simple vent latéral peut déstabiliser l'engin lui donnant ainsi une vitesse latérale. Selon une étude aérodynamique réalisée [6], l'hélicoptère ne devrait pas se déplacer à plus de 1 m/s. Par conséquent, la position devra être déterminée de façon assez rapide pour permettre de repositionner efficacement l'engin. Le taux d'acquisition de la position maximum est fonction de la vitesse de déplacement maximale de l'HPH et de la résolution minimum du système de positionnement choisi.



$$\text{taux d'acquisition max (acqui/s)} = \frac{\text{vitesse max (m/s)}}{\text{résolution (m/acqui)}} \quad (4.1)$$

Ainsi, pour un taux d'acquisition de 10 échantillons par seconde, la technique de détection de la cible devra être sensible à un déplacement de 10 centimètres par acquisition.

#### 4.2 Méthodes pour évaluer la position relative d'un objet

Une revue de la littérature nous a permis de relever plusieurs méthodes pour déterminer la position d'une cible. Bien que plusieurs alternatives efficaces s'offrent à nous, les contraintes suivantes vont limiter le choix optimal pour notre application :

- Le poids du système devant être intégré à l'hélicoptère.
- La puissance de calcul requise pour obtenir la position.
- La résolution et l'erreur relative de la position calculée.
- Le fonctionnement du système dans un environnement intérieur.

La première contrainte fait référence au poids de l'HPH. Puisque la seule puissance disponible est la puissance humaine, il est primordial de minimiser le poids de façon optimale. En second lieu, nous avons vu au deuxième chapitre que l'hélicoptère est muni d'un ordinateur de bord à base d'un microcontrôleur MC68F333, ce qui limite la puissance de calcul disponible pour un système de positionnement qui serait intégré à l'engin. Troisièmement, comme discuté précédemment, la résolution offerte par la méthode de détection de cible joue un rôle important pour nous permettre d'atteindre une vitesse d'échantillonnage adéquate lorsque l'appareil se déplace à sa vitesse maximale.

Enfin, l'hélicoptère exécute tous ses vols dans un environnement intérieur, soit au Stade Olympique de Montréal. Par conséquent, la méthode choisie, pour détecter une cible, doit être parfaitement fonctionnelle dans ce type d'environnement. Avec ces critères de design en tête, nous allons revoir et analyser les caractéristiques de trois méthodes de positionnement pouvant se prêter judicieusement à notre application.

#### 4.2.1 Positionnement avec intervention humaine

Bien qu'il soit spécifié dans les règlements que l'hélicoptère doit limiter ses déplacements à l'intérieur d'une surface déterminée, aucune règle n'impose un système de positionnement autonome. Il est alors possible de développer un système de contrôle de la position en créant une boucle de rétroaction impliquant la vision et la manipulation d'un humain. Le premier concept élaboré s'appuie sur le positionnement d'un hélicoptère traditionnel. Il consiste donc en l'utilisation de la vue d'une personne pour détecter un changement de position de l'appareil. Cette même personne utilise un manche à balai relié à un ordinateur pour communiquer le changement de position. Le pédaleur de l'HPH semble être la personne désignée pour remplir cette tâche comme le serait le pilote d'un hélicoptère traditionnel. Cependant, pour développer les 448 Watts nécessaires pour maintenir l'engin en vol, le pédaleur doit fournir un effort considérable et conserver une concentration parfaite ne lui permettant pas de bénéficier de la stabilité et de la précision requise pour manipuler une forme quelconque de manche à balai.

Le deuxième concept étudié attribue la tâche d'observation et de contrôle à une personne stationnaire à la base de contrôle du vol et non au pédaleur à bord de l'HPH. De cette façon, l'individu observe l'hélicoptère et actionne un périphérique de contrôle relié à un ordinateur. Ce dernier compense les mouvements exécutés par l'opérateur et transmet, par le biais d'un modem radio fréquence (RF), une correction de position à

l'hélicoptère. Cette deuxième approche a comme principal avantage de n'ajouter aucun poids à l'appareil et d'utiliser un ordinateur indépendant pour l'acquisition du périphérique de contrôle, ménageant ainsi les ressources de l'ordinateur de bord. Toutefois, l'opérateur d'un tel système devra nécessairement être situé à plus de 20 mètres du centre de l'hélicoptère, puisque le rayon du rotor de ce dernier est de 18 mètres. De plus, cette personne aura une vue latérale de l'appareil. Par conséquent, il sera difficile d'obtenir une bonne résolution quant au déplacement, surtout lorsque celui-ci sera dans l'axe de profondeur par rapport à la vision de l'opérateur.

#### 4.2.2 Global Positioning System

Une approche très utilisée dans les applications de positionnement est bien entendue le "Global Positioning System" (GPS) ou le "Differential Global Positioning System" (DGPS) [1]. Que ce soit en aviation ou dans le domaine automobile, la position exacte du récepteur GPS peut être déterminée de façon relativement précise ( $< 1$  mètre) et rapide (5 fois/sec.) ce qui en fait un système très approprié pour notre application. Toutefois, deux inconvénients majeurs vont rapidement mettre cette alternative au rancard. Premièrement, l'intégration d'un module récepteur GPS sur l'hélicoptère entraîne nécessairement une pénalité de poids de l'ordre de 100 grammes. Deuxièmement, la portée d'un récepteur GPS limite l'utilisation à l'extérieur des bâtiments. Puisque les vols d'Hélios auront lieu sur l'aire de jeux du Stade Olympique de Montréal, cette méthode est mise de côté.

### 4.2.3 Accéléromètres

Une troisième alternative pour déterminer la position de l'hélicoptère est de mesurer en temps réel l'accélération dynamique de l'appareil sur chacun des axes et ainsi obtenir le déplacement de ce dernier avec la relation suivante:

$$x_{t_2} = x_{t_1} + \int_{t_1}^{t_2} \int_{t_1}^{t_2} a(t) \cdot dt \quad (4.2)$$

Dans cette équation, "x" représente la distance, "t" le temps et "a" l'accélération mesurée.

L'inconvénient majeur de cette technique est l'erreur de précision de la distance accumulée avec les intégrations successives. Il est ainsi primordial de minimiser cette erreur pour que cette technique soit viable. Une façon efficace de mesurer l'accélération est l'utilisation d'accéléromètres.

La compagnie Analog Devices fabrique l'accéléromètre ADXL202. Ce circuit intégré est composé de deux senseurs pour mesurer l'accélération sur deux axes perpendiculaires. De plus, il offre des signaux de sorties numériques pour faciliter son interface avec un microcontrôleur. Ne pesant que quelques grammes, cet accéléromètre pourrait être facilement interfacé à l'ordinateur de bord de l'hélicoptère et mérite donc une analyse plus détaillée sur ses conditions d'utilisation et sur ses performances.

Comme on peut le voir à la figure 4.2, l'ADXL200 est muni de deux sorties numériques que l'on peut interfacer directement avec les entrées TPU du microcontrôleur MC68F333. Ces sorties génèrent une forme d'onde carrée, pour chaque axe de sensibilité, dont le cycle actif est proportionnel à l'accélération ressentie par le circuit

(figure 4.3). L'interface TPU du MC68F333 offre une fonctionnalité microprogrammée pour calculer de façon quasi automatique le "duty cyle" d'une forme d'onde présentée à son entrée. En plus de ne nécessiter que quelques composants externes, ce circuit intégré s'alimente directement à la source de 5 volts.

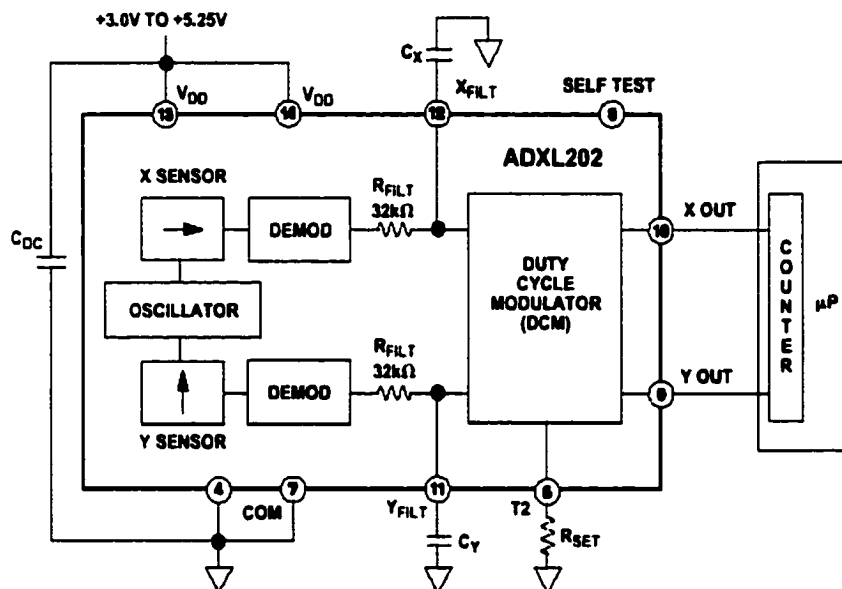


Figure 4.2 Diagramme bloc de l'accéléromètre ADXL200

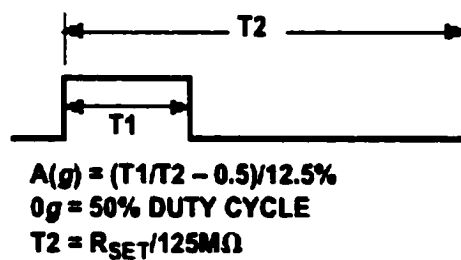


Figure 4.3 Chronogramme de la sortie numérique de l'ADXL200

Par conséquent, l'ADXL202 s'intègre très bien à l'architecture actuelle du module de contrôle à bord de l'hélicoptère.

Du point de vue des performances, l'ADXL202 offre une résolution de 5 mg, soit une accélération minimum de  $49 \text{ cm/s}^2$ . Cette spécification vient assombrir le tableau presque idéal de cette méthode. Toujours selon l'étude aérodynamique réalisée [6], l'hélicoptère peut subir des accélérations presque nulles ( $\approx 0 \text{ cm/s}^2$ ). Par conséquent, l'accéléromètre ne détectera pas les petites accélérations laissant ainsi l'erreur de position augmenter de façon critique. Toutefois, même s'il était possible de détecter une accélération plus petite, les micros accélérations générées par les vibrations de l'appareil produiraient un bruit non négligeable et difficile à éliminer, faussant ainsi les lectures. Par conséquent, le calcul d'une position relative engendre beaucoup trop d'imprécision et nous devons donc maintenant écarter cette méthode de positionnement.

#### 4.2.4 Vision par ordinateur

Une dernière alternative pour détecter un objet est l'utilisation de la vision par ordinateur. Cette méthode, très utilisée dans le domaine de la robotique, bénéficie de plusieurs avantages pouvant se prêter à notre application. Dans un premier temps, l'utilisation d'une caméra pour repérer un objet dans l'espace permet d'obtenir un vaste choix de résolution pour le calcul de la position. Cette dernière dépend de la résolution de la caméra, en pixels, ainsi que la largeur du champ de vision désiré. D'autre part, cette méthode nous permet de calculer la position absolue de la cible à chaque itération, ce qui minimise l'accumulation d'erreur.

Pour minimiser le poids à ajouter à l'hélicoptère, la caméra devra être positionnée au sol de façon à observer la surface de vol. Une référence sera identifiée sur

l'appareil et servira de cible à détecter sur les images générées. Un ordinateur de type Pc s'acquittera des tâches d'acquisition et de calcul de la position. Une correction de position pourra donc être transmise à l'aéronef par l'intermédiaire d'un modem RF. Un autre point intéressant est la possibilité d'utiliser plusieurs caméras pour obtenir un maximum de résolution sur chacun des axes de déplacement de l'hélicoptère. Ainsi, une configuration minimale comporterait une caméra observant l'axe X et une caméra observant l'axe Y. De plus, chaque caméra nous permet d'obtenir une approximation de la hauteur de l'appareil, soit l'axe Z.

Voici un tableau résumé des différentes méthodes de détection de cibles analysées et de leurs principales caractéristiques:

Tableau 4.1  
Résumé des méthodes d'évaluation de la position

	<b>Contrôle Humain</b>	<b>DGPS</b>	<b>Accéléromètres</b>	<b>Vision</b>
Poids ajouté à l'HPH	aucun <sup>1</sup>	≈ 100g	5g	aucun <sup>1</sup>
Interface à l'ordinateur de bord	Aucun <sup>1</sup>	RS-232	TPU	Aucun <sup>1</sup>
Résolution	erreur humaine	< 1m	≈ 49 cm/s <sup>2</sup>	≈ 1,5 cm
Type de position	observation	absolue	relative	absolue

Par conséquent, l'évaluation de la position d'un objet par l'entremise d'un système de vision par ordinateur offre plusieurs avantages et comble toutes les contraintes précédemment énoncées pour notre application. C'est pourquoi un système

---

<sup>1</sup> Il est à noter ici que ces méthodes de détections de cibles utilise le lien RF déjà présent entre la station de vol et l'hélicoptère pour la télémessure. Ainsi aucun composant n'est ajouté à l'appareil.

de positionnement utilisant la vision par ordinateur a été conçu pour détecter et calculer la position d'un HPH en temps réel. Ce système autonome et d'une grande précision fait l'objet d'une description détaillée dans les prochaines sections de ce chapitre.

### 4.3 Configuration du système

Pour être en mesure de calculer de façon précise la position de l'hélicoptère à l'intérieur de la surface de 100 mètres carrés, deux caméras de style "WebCam", offrant une résolution maximale de 640 pixels de largeur par 480 pixels de hauteur, seront utilisées pour mesurer le déplacement sur les deux axes. Voici un schéma illustrant la disposition des caméras:

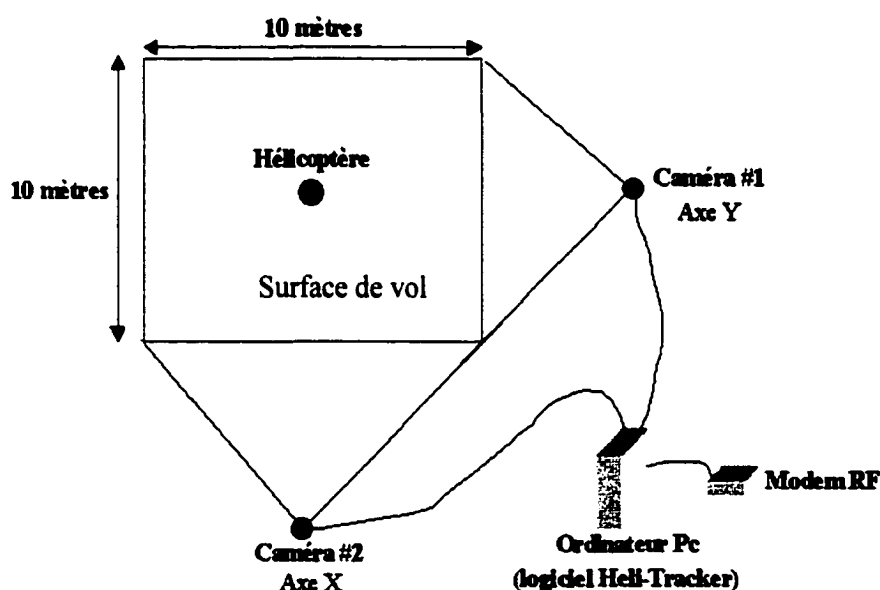


Figure 4.4 Disposition des caméras



Sur ce schéma, on peut voir les deux caméras reliées par un lien USB à un ordinateur Pc. L'orientation des caméras permet d'obtenir une image distincte pour chaque axe de déplacement de l'appareil. Pour permettre une identification efficace et rapide, l'hélicoptère sera muni d'une cible de couleur très contrastée avec l'environnement. De cette façon, les algorithmes de filtrage et de détection de cible seront simplifiés, permettant une exécution beaucoup plus rapide du programme. Il sera possible d'estimer de façon précise le déplacement de l'hélicoptère en calculant le nombre de pixel séparant le centre de l'image au centre de la cible. La figure suivante illustre bien les images filtrées recherchées:

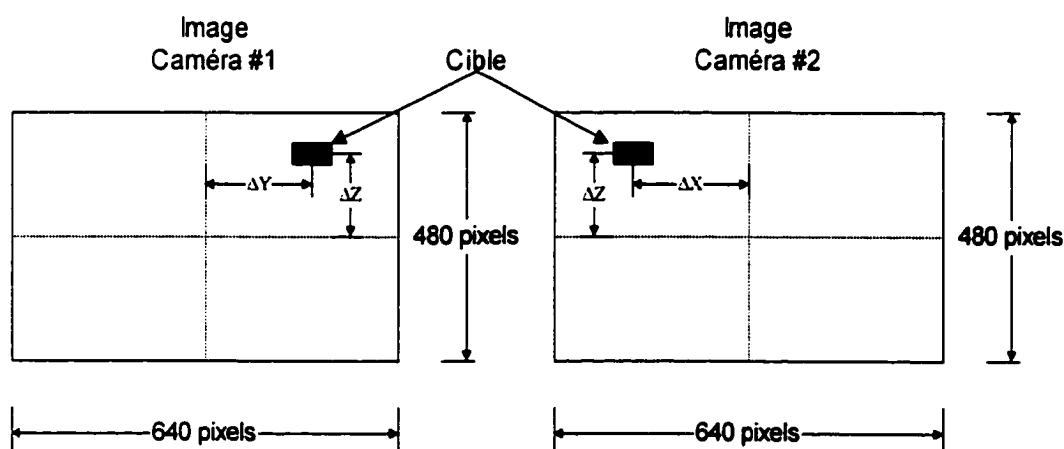


Figure 4.5 Images filtrées obtenues avec les caméras

Sur cette figure,  $\Delta X$ ,  $\Delta Y$  et  $\Delta Z$  représentent respectivement les déplacements selon l'axe X, l'axe Y et l'axe Z. Il est à noter que la mesure selon l'axe Z est en fait la hauteur de l'hélicoptère et n'a aucune influence sur le système de positionnement. On peut remarquer aussi la cible, représentée par un rectangle bleu. Cette dernière est en fait une marque ou une forme fixe sur l'hélicoptère. La problématique de ce système de positionnement réside dans le filtrage des images obtenues avec les caméras, pour isoler

de façon précise cette cible et éventuellement calculer la position. C'est pourquoi un ordinateur Pc est utilisé pour exécuter le logiciel Héli-Tracker, développé dans le cadre de ce projet, pour faire l'acquisition des images et le traitement de ces dernières pour repérer la cible et ainsi calculer la position de l'hélicoptère. Une fois la position déterminée, une correction sera envoyée à l'appareil par le biais d'un modem RF.

Le développement du logiciel Héli-Tracker nécessite l'utilisation de techniques de vision par ordinateur. Ce dernier doit comporter les caractéristiques suivantes pour être en mesure d'offrir une bonne précision et une rapidité acceptable:

- Acquisition rapide des images
- Traitement rapide et efficace des images
- Détection de faibles déplacements de l'hélicoptère

Le logiciel Héli-Tracker a été conçu avec ces objectifs de design en tête. Toutefois, certains choix effectués lors de la conception sont justifiés par des contraintes de coûts et de temps de développement.

#### 4.4 Logiciel de vision "Héli-Tracker"

Le logiciel de vision Héli-Tracker est le cœur du fonctionnement de ce système de positionnement. C'est pourquoi une description détaillée de sa conception et de son fonctionnement est essentielle pour la bonne compréhension de l'ensemble du système de positionnement. Dans un premier temps, un organigramme fonctionnel du système de positionnement a été élaboré, figure 4.6, pour identifier clairement les besoins de notre logiciel et déterminer les critères de design.

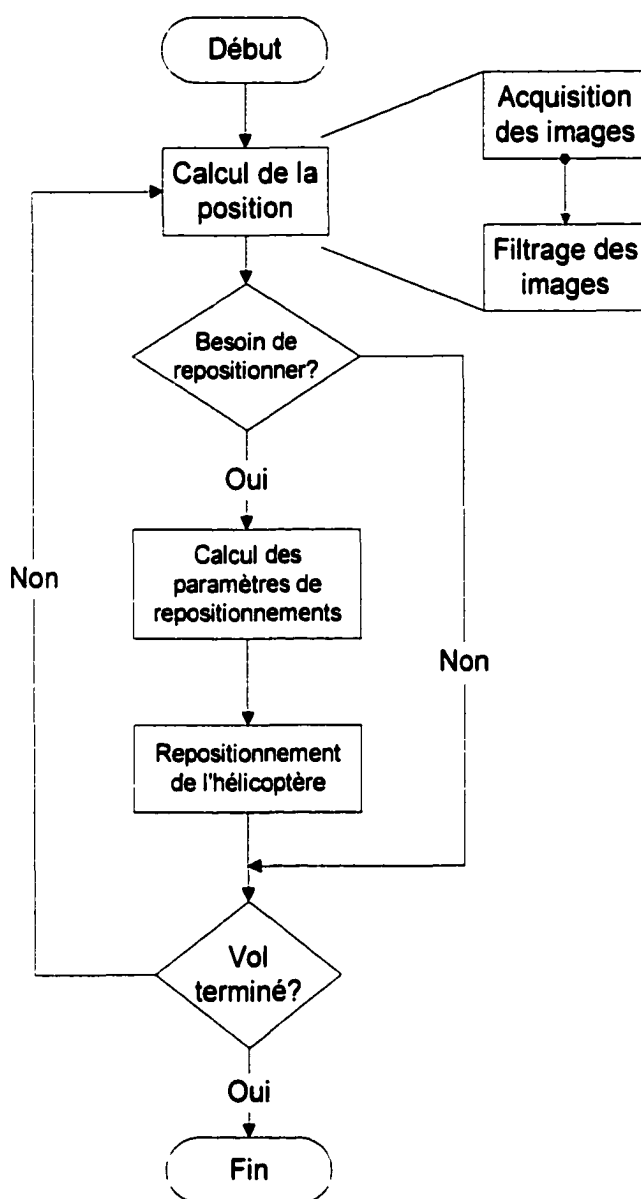


Figure 4.6 Organigramme fonctionnel du système de positionnement

Comme on peut le remarquer à la figure 4.6, l'essentiel du logiciel Héli-Tracker réside dans la conception d'un programme pour calculer la position. Ce dernier est composé de deux sections distinctes: l'acquisition des images et le filtrage de celles-ci. Un premier critère de design est bien entendu le temps d'exécution de ce programme pour nous

permettre d'obtenir une mise à jour de la position de l'HPH en temps réel. Le débit d'images générées par les caméras "WebCam" est spécifié à 30 images par seconde, ce qui est largement supérieur au taux minimum acceptable de 10 échantillons par seconde requis. Par conséquent, le goulot d'étranglement ou "bottleneck" réside dans le filtrage des images. C'est pourquoi, pour le présent projet, un générateur d'images a été conçu pour simuler la portion acquisition réelle des images. En plus de minimiser l'équipement nécessaire pour tester le module de filtrage, cette approche nous permet de créer des images synthèses bruitées pour tester les limites de détection de notre logiciel.

#### 4.4.1 Choix du langage de programmation

L'étape préliminaire au développement du logiciel Héli-Tracker fut le choix d'une plate-forme de programmation sur Pc. Une approche fonctionnelle et efficace est l'utilisation du langage de programmation en C. Ce langage rallie les performances requises pour l'exécution en temps réel à la flexibilité de programmation. Toutefois, le filtrage des images requiert l'utilisation de plusieurs algorithmes de traitement d'images, qui devront nécessairement être programmés et testés en langage C, augmentant ainsi de façon significative le temps de développement du logiciel Héli-Tracker.

C'est pourquoi une autre approche est proposée, soit l'utilisation du logiciel de simulation Matlab®. Ce dernier met à notre disposition toute une librairie d'algorithmes de traitement d'images déjà programmés, permettant ainsi de réaliser des économies considérables de temps, de développement et de tests. De plus, Matlab® est un logiciel mathématique optimisant l'utilisation des matrices. Cette fonctionnalité assurera une gestion efficace des matrices de pixels générées par les caméras. Cependant, l'inconvénient majeur de Matlab® est que le programme conçu est interprété lors de l'exécution et non préalablement compilé. Ceci affecte de façon directe les performances

en temps réel de notre application. Néanmoins, Matlab® est pourvu d'un module "Real Time Workshop" pour les applications exigeantes l'utilisation du temps réel. Ceci nous offre donc la possibilité de convertir éventuellement le programme en langage C pour le compiler avant l'exécution.

#### 4.4.2 Stratégie de développement

La conception du programme Héli-Tracker est segmentée en trois modules: un module générateur d'images, un module détection de la cible et un module de réduction du champ de vision. Cette division est d'autant plus simplifiée avec l'utilisation de fichiers sources séparés de Matlab®. Ainsi, le programme Héli-Tracker est divisé de façon modulaire, le rendant flexible pour toutes modifications ou améliorations. De plus la programmation des différents modules a pu être répartie de façon efficace entre les programmeurs. Voici un schéma illustrant cette segmentation ainsi que les

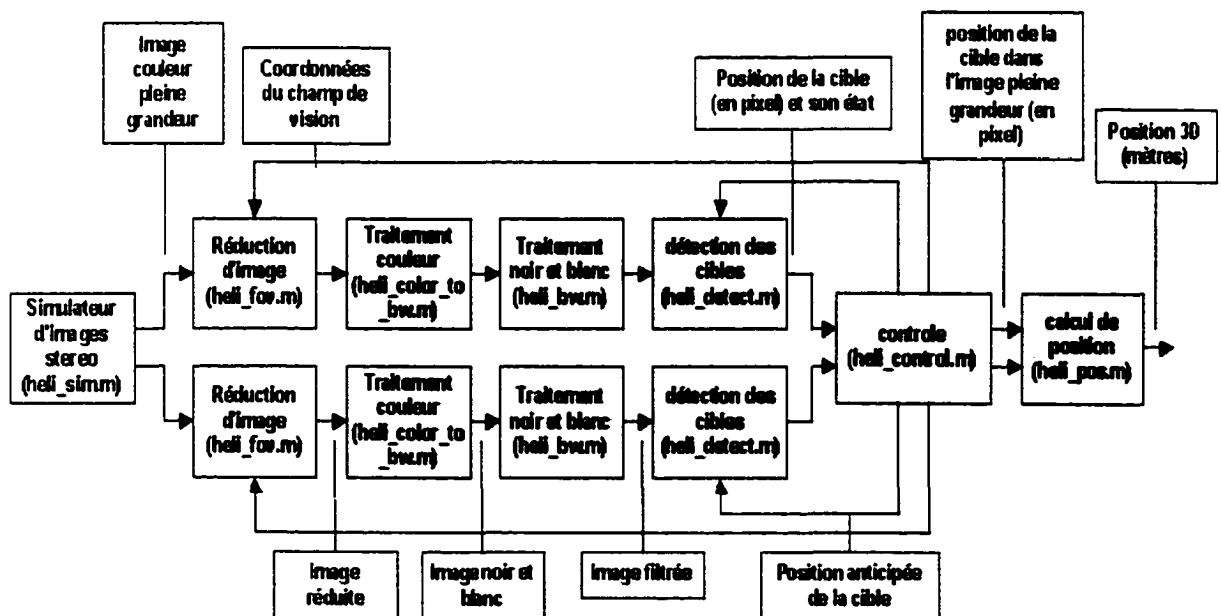


Figure 4.7 Segmentation du logiciel Héli-Tracker en plusieurs fichiers

On remarque sur ce schéma la symétrie qui existe dans le traitement des deux images générées par les caméras. Cette symétrie sera exploitée au niveau des programmes créés, en utilisant les mêmes fonctions ou fichiers pour le traitement associé à chaque image. Voici sous forme de tableau la répartition ainsi que la description sommaire des différents fichiers:

**Tableau 4.2**  
**Segmentation du programme Héli-Tracker**

<b>Modules</b>	<b>Fichiers</b>	<b>Description</b>
Générateur d'images	heli_sim.m	Programme simulateur qui génère deux images avec une cible en mouvement
Détection de la cible	heli_color_to_bw.m	Conversion d'une image couleur en image de niveaux de gris.
	heli_bw.m	Augmentation du contraste et réduction du bruit de l'image.
	heli_detect.m	Détection et identification de la cible.
	heli_pos.m	Calcul de la position 3D.
Réduction de l'image	heli_fov.m	Réduction de l'image.
	heli_control.m	Détermination du champ de vision.

Une autre caractéristique intéressante de la conception modulaire en différents fichiers est la possibilité de faire une analyse détaillée des performances du logiciel. En effet, Matlab® est muni d'une fonction "profile report" permettant d'obtenir le temps d'exécution de chaque fichier. De cette façon, il est possible de repérer plus rapidement le goulot d'étranglement dans l'algorithme et d'apporter des modifications s'il y a lieu.

#### 4.4.3 Module générateur d'images

Ce premier module, constitué d'un seul fichier, est en fait un outil pour valider et tester le fonctionnement des deux autres. Ainsi, il sera remplacé par un module d'acquisition des images dans la version finale du logiciel Héli-Tracker.

Le principal objectif de ce module est de générer deux images représentant la caméra X et la caméra Y, sur lesquelles on retrouve une cible fictive. Pour les besoins de notre simulateur, la cible ajoutée aux images est un rectangle bleu de dimensions fixes (voir figure 4.5). Ce rectangle est en fait la représentation en deux dimensions d'un anneau bleu, placé à l'entour de l'hélicoptère. D'autre part, ce programme sera aussi responsable de déplacer la cible à chaque image générée selon une séquence pré-établie.

Pour en faire un simulateur d'images efficace représentant le plus possible les conditions réelles d'utilisation, ce module est pourvu de paramètres d'entrées permettant d'apporter des variations sur les images générées. Dans un premier temps, il est possible d'utiliser un fichier image de format JPEG pour créer l'arrière plan de chaque image générée. De cette façon, il est possible d'utiliser une photographie illustrant le mieux possible le décor réel. Sur cet arrière plan est ajoutée une cible rectangulaire bleue d'intensité variable. Nous pourrions vérifier ainsi l'intensité minimum requise pour être en mesure d'identifier la cible. Une autre fonctionnalité est le scintillement de la cible. Cette option permet de faire disparaître la cible momentanément de l'image pour simuler la situation où cette dernière n'est pas identifiée par l'algorithme de filtrage. Plusieurs raisons peuvent provoquer cette situation, entre autre si la pale de l'hélicoptère obstrue momentanément le champ de vision de l'une des caméras. De plus, nous verrons plus loin dans ce chapitre que cette fonctionnalité permettra de valider le fonctionnement de l'algorithme de réduction de l'image. Un quatrième paramètre d'entrée permet de tester la

robustesse de l'algorithme de filtrage en ajoutant du bruit aux images générées. Ce dernier est de type "poivre et sel", c'est-à-dire qu'un certain nombre de pixels, choisis au hasard, ont changé de couleurs pour ainsi brouter les images. La densité de ce bruit peut être ajustée par l'intermédiaire de ce paramètre. Finalement, une dernière variable d'entrée à ce module est bien entendu le temps. De cette façon, le générateur d'images peut faire avancer la cible selon l'instant où est rendue la simulation. Voici un tableau résumant ces différents paramètres :

**Tableau 4.3**  
**Paramètres d'entrée du générateur d'images**

<b>Paramètres</b>	<b>Description</b>
Choix de l'arrière plan image X et Y	Spécifie deux fichiers image de format JPEG pour servir d'arrière plan aux deux images synthétiques produites.
Bruit ajouté	Spécifie la densité du bruit de type "poivre et sel" à ajouter aux images (0.01 à 0.05). 0 $\Rightarrow$ aucun bruit
Scintillement	Active l'option de scintillement pour faire disparaître la cible sur certaines images.
Intensité de la cible	Spécifie l'intensité de la cible bleu (0 à 255).
Temps	Temps de la simulation du déplacement de la cible.

Une fois ce programme exécuté, deux images sont générées en mémoire sous forme de deux matrices à trois dimensions chacune. En effet, l'acquisition d'images en couleurs est représentée par une matrice comportant une dimension pour chaque couleur: rouge, vert et bleu. Chacune de ces matrices est de grandeur égale à la résolution de la caméra, soit 640 colonnes par 480 lignes pour notre cas.



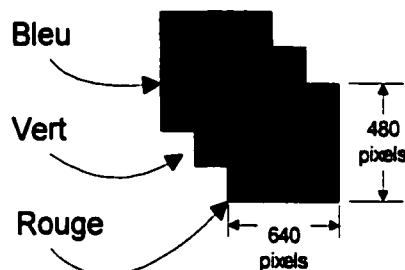


Figure 4.8 Représentation matricielle d'une image

#### 4.4.4 Module détection de la cible

Le module de détection de la cible est en fait le cœur du système de positionnement. En effet, c'est celui-ci qui est responsable du filtrage des images et de l'identification de la cible pour permettre d'obtenir éventuellement les coordonnées en trois dimensions de celle-ci donc de l'hélicoptère. Étant donné que les procédés de filtrage sont exactement les mêmes pour les deux caméras, nous concentrerons cette discussion simplement sur l'analyse d'une seule image. La corrélation des deux images pour en obtenir la position exacte de la cible sera décrite dans la section suivante. Par conséquent, chaque image traversera cinq étapes de traitement distinct.

##### 4.4.4.1 Conversion de l'image couleur en niveau de gris

Ce module reçoit en entrée une matrice de dimensions  $640 \times 480 \times 3$ . La démarche adoptée pour isoler la cible du reste de l'image est la suivante. Dans un premier temps, l'image couleur est convertie en image à intensité de niveaux de gris. Étant donné que la cible est bleue, la première technique de conversion que nous avons implémentée consiste simplement à extraire de l'image couleur la composante bleue et de la

considérer comme une image d'intensité. Bien que cette technique soit suffisamment efficace pour une image idéale, elle ne l'est définitivement pas pour une image réelle. Le principal problème est qu'une image réelle peut contenir plusieurs objets avec un blanc très intense ce qui amène beaucoup d'objets non-désirés dans la composante de bleu. Ceci ralentit le temps de calcul global et peut amener à une éventuelle mauvaise détection de cible. Par conséquent, un autre algorithme plus efficace a été proposé. On extrait toujours la composante de bleu de l'image couleur mais on met à zéro tous les pixels de celle-ci dont la composante rouge ou verte est plus grande qu'un certain seuil préalablement défini. Ceci enlève donc de l'image tous les pixels avec de fortes composantes de rouge et de vert, dont le blanc qui lui a une valeur très élevée de rouge, vert et bleu.

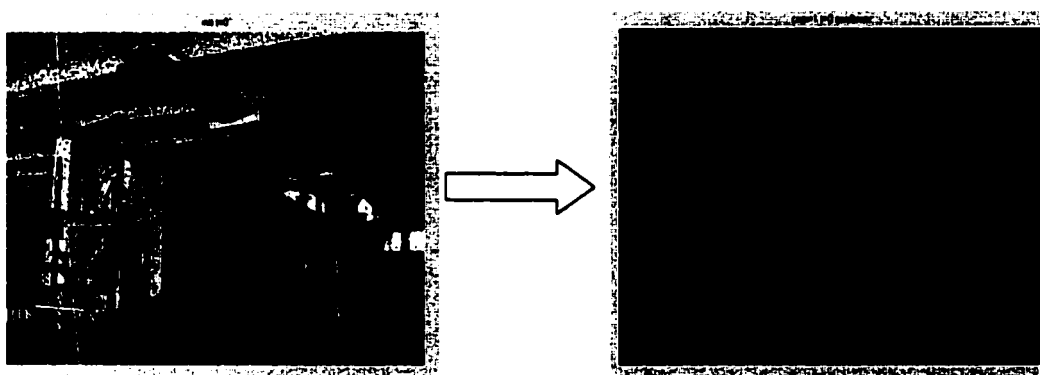


Figure 4.9 Conversion couleur à niveaux de gris

#### 4.4.4.2 Augmentation du contraste de l'image

La matrice de deux dimensions résultantes est ensuite contrastée à l'aide d'une fonction disponible avec la boîte d'outils de traitement d'images de Matlab© (imadjust).

Le contraste de l'image est ainsi augmenté en appliquant un seuil à l'image et en ré-étalant les pixels restants entre 0 et 255 à l'aide de la fonction linéaire suivante :

$$P = (P - \min) * 255 / (\max - \min)$$

$P$  = tous les pixels de l'image

$\min$  = valeur minimum de tous les pixels de l'image

$\max$  = valeur maximum de tous les pixels de l'image

Cette conversion permet d'éliminer les pixels de faibles intensités de l'image source et amènera un seuillage plus sélectif lors de l'étape de binarisation qui suivra.

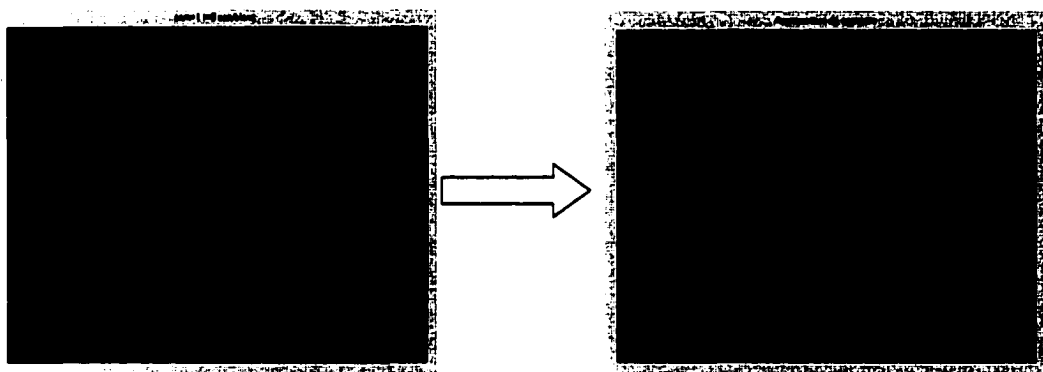


Figure 4.10 Augmentation du contraste de l'image

#### 4.4.4.3 Réduction du bruit

Maintenant que la plage dynamique a été considérablement augmentée, la prochaine étape de traitement consiste à appliquer un filtre qui permettra d'enlever tous les petits points provenant de petits objets ou du bruit de l'image. Nous avons appliqué un filtre médian avec un masque matriciel de 4×4 à l'image d'entrée pour éliminer tout le bruit ponctuel de cette dernière. Encore une fois, la librairie de Matlab© met à notre disposition une fonction pour exécuter ce filtre médian (`medfilt2`).

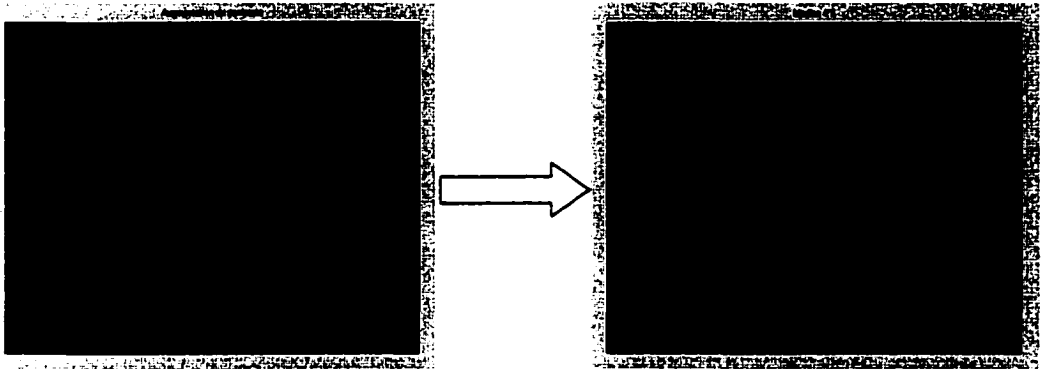


Figure 4.11 Filtrage de l'image avec un filtre médian

#### 4.4.4.4 Conversion en image logique

Maintenant qu'un filtrage suffisant a été appliqué à l'image, il ne reste qu'à la convertir en image binaire à 2 niveaux de gris (0 ou 1) afin de pouvoir y extraire le ou les objets qui sont susceptibles d'être la cible recherchée. Pour cette étape, nous avons défini statistiquement un seuil de 30 qui sera utilisé pour définir la valeur des pixels. Ainsi, toutes les valeurs de pixels plus grandes que 30 deviendront égales à 1 et les autres se verront attribués une valeur de 0.

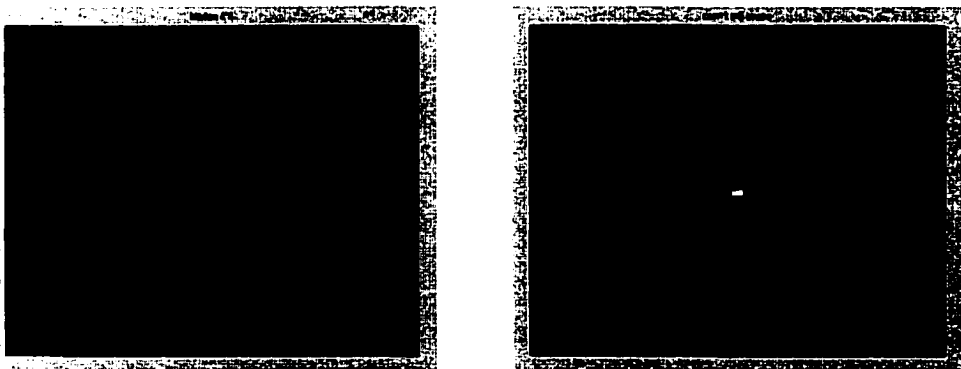


Figure 4.12 Conversion de l'image en image logique binaire

#### 4.4.4.5 Détection des objets et sélection de la cible

Suite à l'obtention d'une image binaire, le programme doit déterminer tous les agglomérats de pixels dont la valeur est 1. Chaque agglomérat deviendra un objet qui serait susceptible d'être la cible recherchée. Matlab© fournit plusieurs outils intéressants pour cette étape. La fonction "bwlabel()" retourne une nouvelle image où tous les objets trouvés ont été étiquetés séquentiellement. Le nombre d'objets trouvés est aussi retourné. De cette nouvelle image, on peut extraire tous les pixels d'un objet à l'aide de la fonction "find()". On peut ensuite utiliser d'autres fonctions comme "bw\_area()" qui retourne l'aire de l'objet ou "bwperim()" qui retourne une image du contour (périmètre) de l'objet.

Dans notre cas, la cible est choisie selon les critères suivants :

- Aire de l'objet : L'aire de la cible doit être comprise entre deux valeurs limites. Ces valeurs ont été définies selon les variations possibles de la cible en fonction de sa distance par rapport à la caméra et selon la variation d'aire due au filtrage et traitement de l'image. Bien que ce dernier aspect ne semble pas important à première vue, il est très important de le considérer sérieusement. En effet, une image fortement bruitée amènera généralement une image plus petite due au moyennage qui fera passer certains pixels sous la valeur de seuillage. L'effet inverse est tout aussi possible.

- Déplacement depuis le dernier repérage : La cible est initialement à une position connue de notre système. Pour tous les repérages de cibles subséquents, on rejettera tout objet qui indiquerait un déplacement plus grand qu'une certaine limite de déplacement en X et Y.

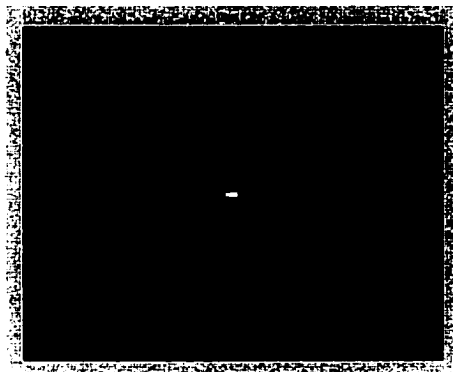


Figure 4.13 Cible détectée

#### 4.4.5 Module réduction de l'image

Dans un contexte de contrôle en temps réel de l'hélicoptère, le temps requis pour traiter les images échantillonnées par les caméras est critique. En effet, le processus de détection de la cible décrit précédemment, implique des calculs et des recherches dans les matrices images. Par conséquent, si ces dernières sont de grandes dimensions, un délai temporel considérable sera ajouté pour calculer la position de l'hélicoptère. Ceci aura nécessairement un effet néfaste sur le temps de réponse et la stabilité globale du système de positionnement. Il devient donc impératif de développer une méthode pour accélérer le traitement des images. Une approche efficace est de réduire la grandeur des matrices images et par le fait même le nombre de pixels à traiter.

##### 4.4.5.1 Réduction de la résolution versus réduction du champ de vision

Une première méthode pour réduire la grandeur des matrices à traiter est de réduire directement la résolution des images échantillonnées. À titre d'exemple, on peut faire passer la résolution des caméras de  $640 \times 480$  pixels à  $160 \times 120$ . Toutefois, bien que

le nombre de pixels à traiter par couleur primaire passe de 307 200 à 19 200 (16 fois moins), la définition de la cible se trouve affaiblie. En effet, cette réduction du nombre de pixels pour définir l'image se répercute aussi sur le nombre de pixels pour définir la cible elle-même. Ainsi, la perte d'information utile aux différents algorithmes de filtrage et de détection, nous amène à rejeter cette première approche.

Une alternative pour réduire le nombre de pixels à traiter est de réduire le champ de vision en segmentant l'image. De cette façon, en diminuant le champ de vision par un facteur de quatre, nous obtenons toujours une réduction du nombre de pixels d'un facteur de 16. Toutefois, la résolution initiale est conservée comparativement à la méthode précédente. La cible est donc toujours définie par le même nombre de pixels et aucune information utile n'est perdue dans ce processus. La réduction du champ de vision a aussi un autre avantage très appréciable : elle élimine toutes fausses cibles se trouvant à l'extérieur du champ de vision. Les fausses cibles sont des éléments de l'arrière plan pouvant être interprétés par le module de détection comme une vraie cible. Ceci a pour effet de confondre la vraie cible et d'obtenir une position erronée. La réduction du champ de vision augmente donc la robustesse du système.

On peut voir sur la figure ci-dessous une comparaison des effets de la réduction de la résolution par rapport à la réduction du champ de vision. Pour les fins de l'exemple, on pose comme hypothèse que le champ de vision initiale des caméras est de 20 mètres de largeur par 15 mètres de hauteur et que la résolution des caméras est ajustée à 640 par 480 pixels. De plus, une cible de 60 centimètres par 30 centimètres est utilisée.

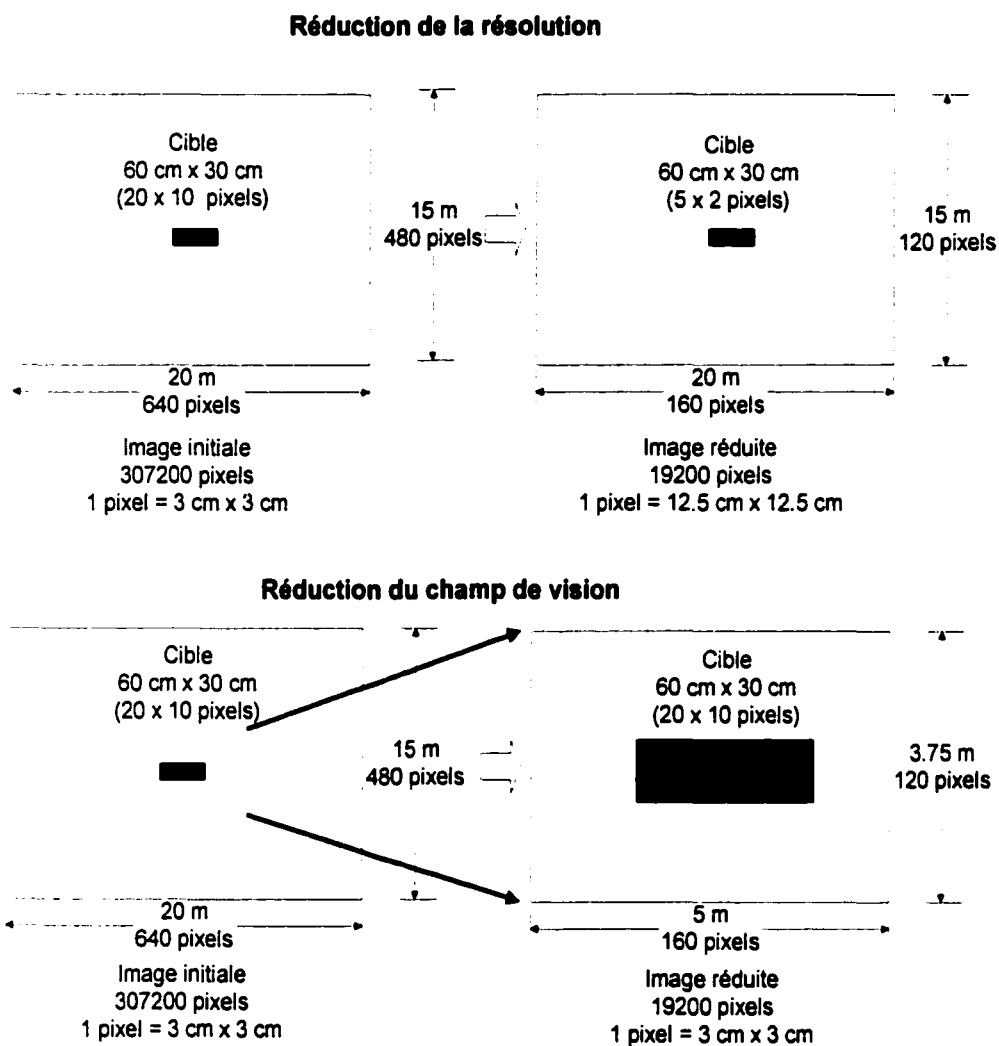


Figure 4.14 Réduction de la résolution versus réduction du champ de vision

Voici maintenant un échantillon d'un champ de vision typique obtenue à partir d'une image générée par le simulateur d'images:



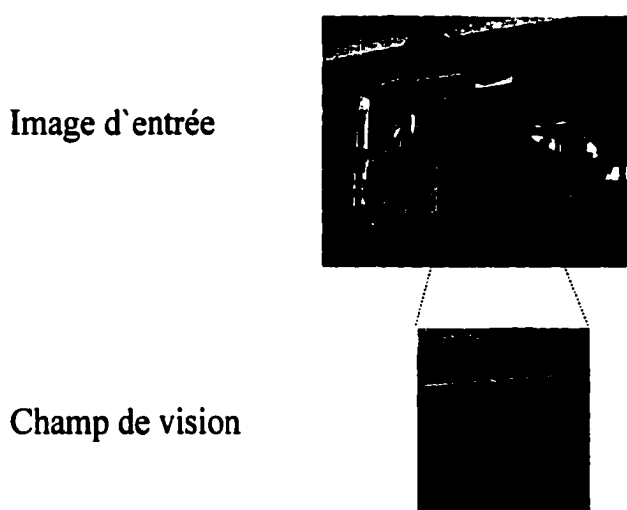


Figure 4.15 Exemple d'un champ de vision avec arrière plan du simulateur d'images

On remarque ici à la figure 15 que pour réduire efficacement le champ de vision, la position de la cible doit être préalablement obtenue. Il faudra donc implanter un algorithme de réduction de l'image en considérant la position de la cible calculée à l'itération précédente. Voici un dernier exemple illustrant l'élimination des fausses cibles par l'intermédiaire de la réduction du champ de vision:

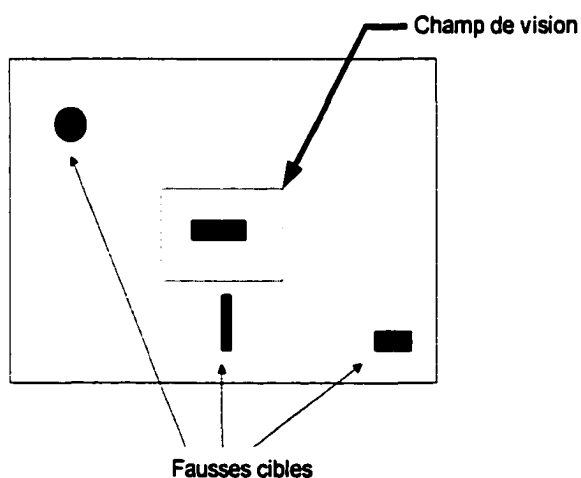


Figure 4.16 Réduction du champ de vision et élimination des fausses cibles

#### 4.4.5.2 Implantation de l'algorithme

L'algorithme de réduction de l'image réside dans deux fichiers, donc deux programmes séquentiels. Dans un premier temps, le programme "heli\_control.m" est responsable de poursuivre la cible afin d'établir la grandeur et la localisation du champ de vision dans l'image originale. Ce processus doit être fait pour chacune des caméras. Les données dont il a besoin sont la position des cibles et l'état de la détection. Ces données sont fournies par le module de détection de cible. L'état de la détection est simplement une valeur booléenne indiquant s'il y a eu détection ou non de la cible. Par la suite, les coordonnées du nouveau champ de vision sont transmises au programme "heli\_fov" qui exécute la réduction du champ de vision (Field Of View) sur la nouvelle image échantillonnée. On retrouve à la figure 17 un schéma bloc explicite du module de réduction de l'image.

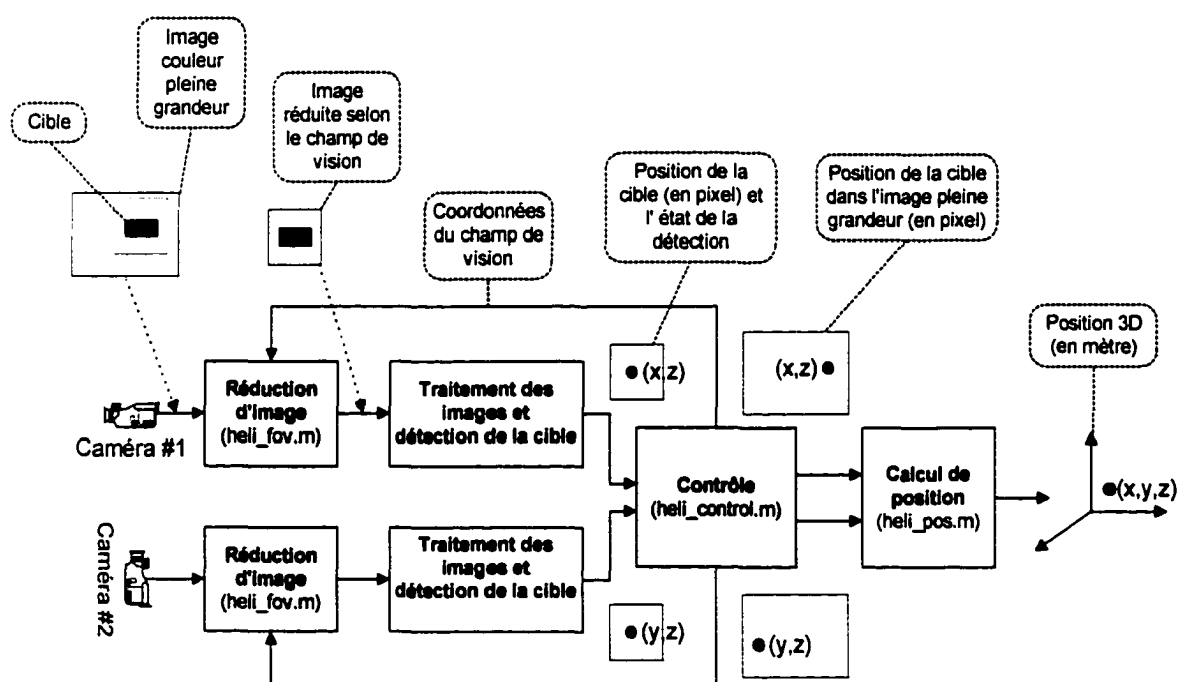


Figure 4.17 Schéma bloc du zonage de l'image

D'autre part, la poursuite de la cible exige un traitement particulier capable de s'assurer d'un cadrage efficace de celle-ci. De plus, la dimension du champ de vision qui vient se resserrer sur la cible peut être de grandeur variable pour optimiser au maximum le nombre de pixels à traiter et du même fait le temps de calcul. C'est pourquoi un algorithme simple de poursuite de cible (target tracking) est utilisé afin de localiser la cible en tout temps.

L'algorithme fonctionne sous deux modes : recherche et poursuite. Le mode recherche désigne l'état où la position de la cible n'est pas connue. Ce mode est sélectionné à l'initialisation du système. Le mode poursuite désigne l'état où la position de la cible est connue. Pour passer du mode recherche au mode poursuite, un nombre minimum ( $D_b$ ) de détections consécutives, signalées par le module de détection, est requis. Ceci a pour but de s'assurer que la bonne cible a été repérée et éviter de poursuivre une fausse cible ponctuelle. Actuellement, les simulations que nous avons exécutées démontrent que  $D_b = 2$  donne de bons résultats. Lorsque l'algorithme est en mode poursuite, il réduit à chaque itération le champ de vision jusqu'à une superficie minimum. Pour passer du mode poursuite au mode recherche, un nombre minimum ( $D_m$ ) de non-détection consécutive, signalée par le module de détection, est aussi requis. Ceci permet d'éviter d'agrandir le champ de vision inutilement lorsque la cible disparaît momentanément à cause du bruit ou pour d'autres raisons. Il est à noter que l'agrandissement du champ de vision entraîne une forte pénalité temporelle pour le traitement de l'image. Actuellement, les simulations que nous avons exécutées démontrent que  $D_m = 1$  ou 2 donnent de bons résultats. En mode poursuite, l'algorithme augmente le champ de vision à chaque itération jusqu'à une superficie maximum correspondant à l'image originale. La figure suivante illustre l'organigramme opérationnel de cet algorithme de poursuite.

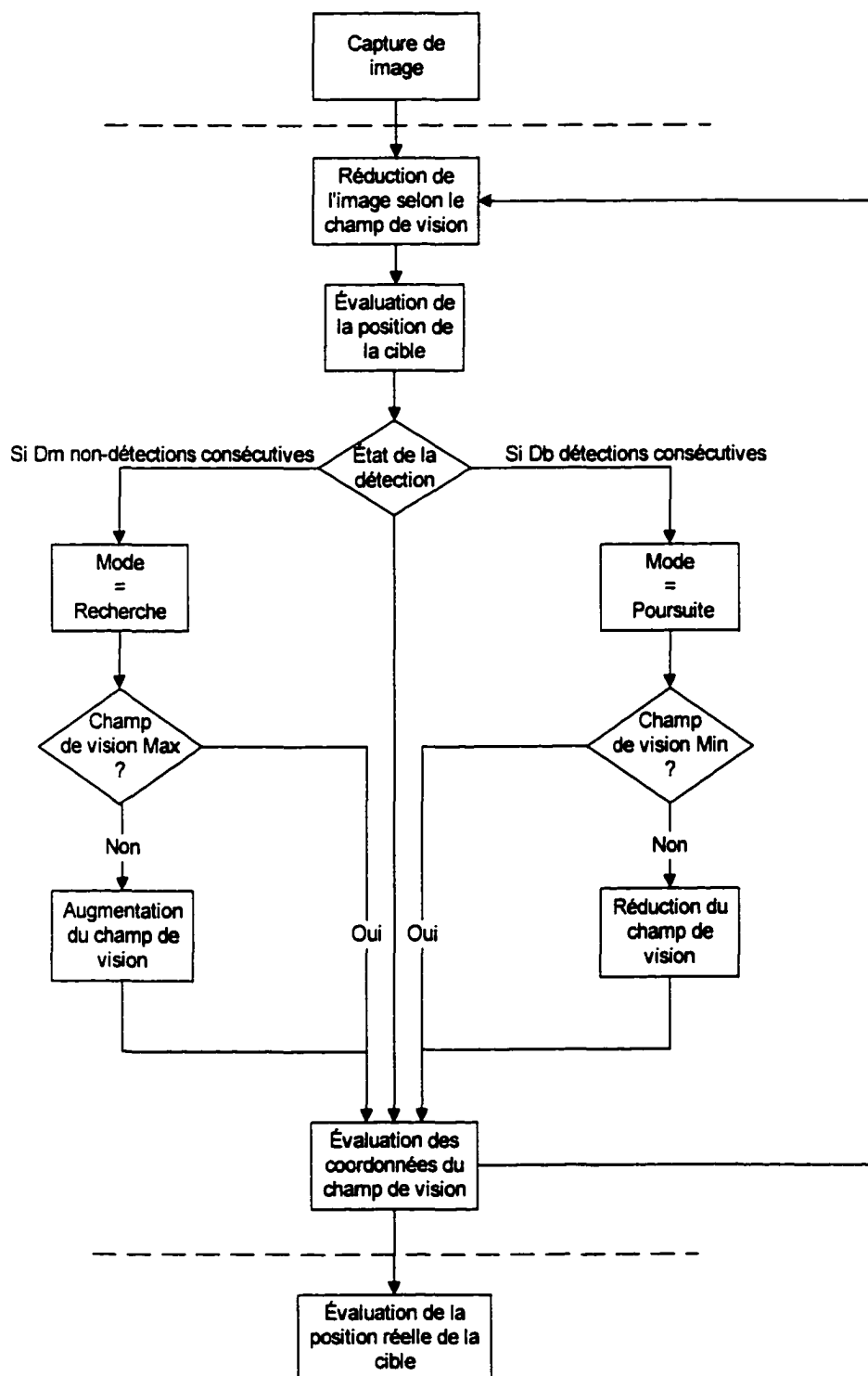


Figure 4.18 Organigramme opérationnel de l'algorithme de poursuite de la cible

#### 4.4.6 Évaluation de la position de la cible

Une fois la cible détectée, les coordonnées du centre de celle-ci ainsi que les coordonnées du champ de vision sont utilisées pour calculer la position en trois dimensions de la cible dans la scène capturée avec les caméras. La figure 19 illustre la relation existant entre les coordonnées en pixels et la position réelle de la cible, selon les trois axes, en mètres.

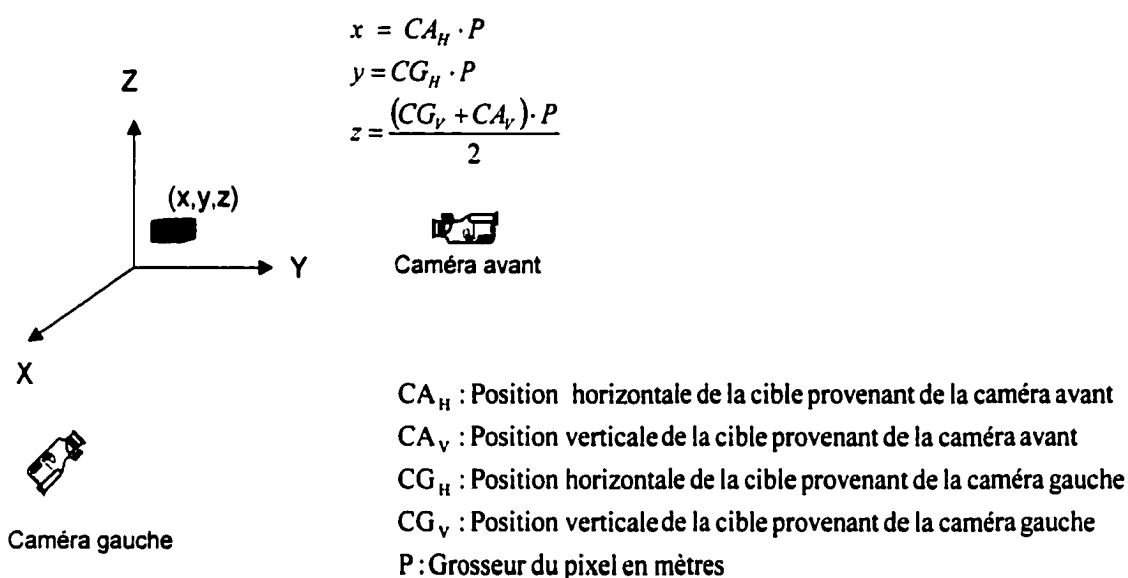


Figure 4.19 Calcul de la position de la cible dans la scène en trois dimensions.

#### 4.4.7 Analyse des performances

Puisque le système de vision devra éventuellement fonctionner en temps réel, nous avons fait une analyse du temps de simulation requis par MATLAB pour exécuter les fonctions principales du programme. Les simulations ont été générées avec MATLAB 5.3 sur un processeur AMD Athlon 600 MHz avec 128 MO de mémoire vive.

Le tableau qui suit montre la durée moyenne pour chacune de ces fonctions pour trois cas différents.

**Tableau 4.4**  
**Temps d'exécution moyen des fonctions**

<b>Nom de la fonction</b>	<b>Description</b>	<b>Temps moyen pour traiter les deux images (ms)</b>		
		<b>Cas #1 Image simple avec réduction du champ de vision</b>	<b>Cas #2 Image simple sans champ de vision</b>	<b>Cas #3 Image complexe avec champ de vision</b>
heli_fov	Réduction de l'image	20		20
heli_color_to_bw	Traitement de la couleur et conversion en noir et blanc	144	555	249
heli_bw	Traitement en noir et blanc	426	1455	664
heli_detect	Détection de la position de la cible	49	49	78
heli_control	Contrôle du champ de vision	0	0	0
heli_pos	Calcul de la position	0	0	0
	<b>Total</b>	<b>639<sup>2</sup></b>	<b>2059<sup>2</sup></b>	<b>1011<sup>2</sup></b>

En comparant les cas 1 et 2 on peut évaluer l'importance de faire une réduction du champ de vision sur l'image afin de limiter le temps d'exécution. En effet, en réduisant les images d'entrée on réduit du tiers le temps d'exécution pour traiter les images des deux caméras.

---

<sup>2</sup> Le temps total de chaque simulation comprend les routines d'affichage graphique qui seront éliminés dans la version finale du programme pour accélérer le traitement.

D'autre part, les cas 1 et 3 montrent que le temps d'exécution a une relation directe avec la complexité des images. La complexité de l'image d'entrée engendre deux choses : détection de plusieurs objets et la perte de la cible. Le fait de détecter plusieurs objets augmente le temps d'exécution, car on doit faire plusieurs opérations récursives sur ceux-ci pour ensuite déterminer quel objet est la véritable cible. La perte de la cible fait en sorte que le champ de vision augmente jusqu'à ce qu'on retrouve la cible. Il est donc évident que le temps d'exécution augmente. Par ailleurs, les sous-programmes `heli_control` et `heli_pos` ont un temps d'exécution relativement court en comparaison aux autres sous-programmes, car ils n'utilisent pas de boucle et ne font pas de traitement de matrice.

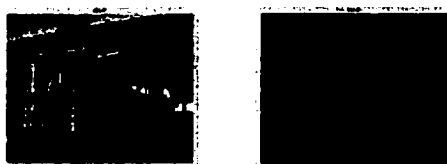


Figure 4.20 Image complexe et simple utilisées pour les tests.

#### 4.5 Repositionnement de l'hélicoptère

Une fois la position de l'HPH déterminée, l'ordinateur sur lequel s'exécute le logiciel Héli-Tracker calcul les paramètres de repositionnement et transmet à l'hélicoptère, par le biais du lien RF, les données nécessaires au changement de position. L'approche utilisée pour le repositionnement de l'appareil ne fait pas l'objet de ce mémoire, mais est à toute fin pratique le complément essentiel du système de positionnement. La méthode envisagée est de modifier le centre de gravité de l'aéronef pour exercer un redressement contrôlé de celui-ci. Pour ce faire, deux alternatives s'offrent à nous: modifier la position du pédaleur ou déplacer deux masses selon des axes perpendiculaires. Pour des raisons de stabilité, le premier choix a dû être rejeté. C'est ainsi qu'à l'aide de deux masses, assemblées sur des rails de façon perpendiculaires, il

sera possible de modifier le centre de gravité de l'HPH. Ces dernières seront contrôlées par deux moteurs de même modèle que ceux utilisés pour l'ajustement du pas des pales. Encore plusieurs paramètres restent à être déterminer, en ce qui a trait aux calculs nécessaires pour modifier le centre de gravité selon la position de l'appareil. Toutefois, les premières études théoriques ont démontré la faisabilité de cette méthode. Des simulations à venir avec notre modèle dynamique [6] vont permettre de préciser l'algorithme ainsi que les différents paramètres impliqués.



## DISCUSSION ET INTERPRÉTATION DES RÉSULTATS

Le projet Hélios n'étant pas encore complété, aucun test n'a été réalisé à ce jour sur le banc d'essais. Les premières simulations sont prévues pour le début de l'année 2001 et permettront de valider plusieurs concepts énoncés. Toutefois, il est important de souligner quelques résultats obtenus. Dans un premier temps, il a été fait mention initialement que la stabilité de l'hélicoptère repose sur l'ajustement des angles d'attaque des pales en temps réel lors du vol. L'utilisation d'un ordinateur de bord permet d'accomplir cette tâche de façon précise. La faible vitesse de rotation des pales ainsi que les contraintes de poids étaient des obstacles majeurs à l'implantation d'un tel système. Mais l'approche de contrôle en boucle ouverte, basée sur ce qui se fait en matière d'injection électronique dans les véhicules automobiles, nous a permis d'obtenir un contrôle à la fois rapide, efficace et léger. L'utilisation d'une table de données par l'ordinateur permet de trouver précisément le pas de chacune des pales.

Il est important de citer ici, que cette approche a toutefois ses limites. En effet, puisque des simulations sont requises pour calibrer l'assemblage pales/rotor, l'hélicoptère est contraint à exécuter des vols préalablement simulés. De plus, le vol doit avoir lieu dans les mêmes conditions que la simulation pour assurer l'exactitude des données tabulées. Ainsi, si une défectuosité est décelée sur l'une des pales et qu'un remplacement est requis, il sera nécessaire de retourner sur le simulateur pour calibrer à nouveau l'assemblage avec la nouvelle pale.

Pour sa part, l'unité de contrôle utilisée nous a permis de sauver beaucoup de temps de développement. En effet ayant déjà été mis à l'épreuve avec le sous-marin

OMER, la fiabilité et la modularité de l'ordinateur de bord se sont avérées des critères essentiels au développement du module de contrôle. De plus, sa conception très compacte et légère assure un poids ajouté minimum à l'HPH.

En ce qui a trait au simulateur, le système d'acquisition développé pour recueillir les données de simulation se devait d'être efficace et convivial. En effet l'assemblage et la mise en fonction du banc d'essais pour exécuter une simulation requièrent le déplacement de celui-ci au Stade Olympique. C'est pourquoi il était nécessaire d'avoir recours à un outil d'acquisition fiable facile de transport. L'utilisation de l'ordinateur de bord comme unité centrale de traitement nous a permis d'obtenir un système d'acquisition facilement intégrable au banc d'essais sous forme d'une boîte de contrôle. De plus le lien RF permet une analyse en temps réel des simulations et l'interface sur plate-forme Pc facilite l'analyse et la sauvegarde des données.

D'autre part, l'hypothèse de stabilité sur laquelle repose toute la conception de l'hélicoptère peut maintenant être appuyée par un système précis de positionnement. En effet le système de vision assisté par ordinateur permet de détecter la position de l'appareil et de permettre d'apporter les ajustements nécessaires. Le succès de cette méthode repose essentiellement sur sa rapidité d'exécution et sa précision de détection.

Les résultats obtenus ont été réalisés à partir d'images synthèses de test. Il va sans dire qu'il faut être prudent dans l'interprétation de ces résultats étant donné l'approximation de la réalité qui est faite. Toutefois, ces tests ont permis de vérifier la robustesse de l'algorithme développé avec divers degrés de complexité de l'image. Nous pouvons retirer de ces essais qu'une cible très contrastée avec l'arrière plan est requise pour accélérer le traitement des images.

De plus, selon les résultats obtenus, la technique de poursuite de la cible implantée permet de diminuer de plus du tiers le temps requis pour déterminer la position de la cible. Ce gain en performance est non négligeable dans un contexte temps réel. Il faut cependant tenir compte ici que la cible était toujours détectée à chaque traitement. Dans le cas réel d'application de cette technique, plusieurs facteurs peuvent faire en sorte que la cible ne soit pas toujours détectée et ainsi provoquer un agrandissement du champ de vision. Le gain en performance sera donc affecté et il est par conséquent plus réaliste de considérer une diminution de moitié du temps de calcul.

Finalement, un autre résultat intéressant associé au système de positionnement par vision est l'absence de composants supplémentaires à ajouter à l'hélicoptère. En effet seulement deux caméras et un ordinateur Pc, disposés sur le périmètre de l'espace de vol, sont requis. De cette façon, aucun poids n'est ajouté à l'appareil permettant ainsi de préserver le critère de légèreté.

## CONCLUSION

Le design de l'hélicoptère à propulsion humaine Hélios repose sur la conception d'un module de contrôle et d'un système de positionnement capable d'assurer la stabilité de l'engin et de maximiser le transfert de puissance entre le pédaleur et les rotors. Pour accomplir ces tâches, un ordinateur de bord est intégré à l'hélicoptère et assure le contrôle en temps réel des angles d'attaque des pales. Cette hypothèse de contrôle en temps réel est rendue possible grâce à l'utilisation d'une table de données permettant, avec un minimum de calculs, d'obtenir rapidement l'angle de chaque pale. Par ailleurs, le choix de l'unité de contrôle devait satisfaire des contraintes d'espace et de poids. C'est pourquoi l'ordinateur développé dans le cadre du sous-marin OMER fût utilisé et adapté au projet Hélios avec l'ajout de différents capteurs.

D'autre part, la génération de la table de données requise par l'algorithme de contrôle nécessite une simulation précise des conditions de vols. C'est pourquoi un banc d'essais fût fabriqué pour permettre de calibrer l'assemblage pales/rotors. L'efficacité de ce simulateur repose sur la structure mécanique et sur le système d'acquisition. L'ordinateur de bord fût utilisé pour permettre une flexibilité de développement et une réutilisation des programmes développés. L'interface d'analyse des données sur Pc, de son côté, répond aux besoins initiaux en matière d'interprétation des résultats. Il est en effet possible de visualiser facilement les données recueillies et de créer la table de données des angles d'attaque.

Finalement, le système de vision par ordinateur répond bien aux objectifs fixés en matière de positionnement. Plusieurs méthodes de détection de cible ont été étudiées,

mais une seule d'entre elles se prêtait à notre application. Il est alors possible de déterminer de façon précise et relativement rapide la position de l'hélicoptère. Cette technique utilise des algorithmes de traitement d'images et une méthode de poursuite de la cible pour accélérer le temps requis pour identifier celle-ci. Le système de positionnement développé n'ajoute aucun poids à l'appareil, respectant les objectifs de légèreté visés.

## RECOMMANDATIONS

Le système de positionnement nécessite l'utilisation du logiciel Matlab pour l'exécution de l'algorithme de traitement des images. Ce logiciel fait une interprétation du programme en cours d'exécution ce qui allonge le temps de calcul de la position. Il est donc recommandé d'étudier la possibilité d'implanter cet algorithme en langage C, pour ainsi pouvoir compiler le programme et accélérer le temps d'exécution. De plus, l'intégration des routines d'acquisition des images à partir des WebCam sera grandement simplifiée.

Par ailleurs, le choix de la cible à positionner sur l'hélicoptère peut être étudié plus en détail. En effet, il est suggéré dans ce rapport de placer un anneau de couleur vive autour de l'appareil, mais plusieurs autres approches permettant d'avoir une cible très contrastée avec l'arrière plan sont possibles. Il est conseillé entre autre d'utiliser de petites lumières de forte intensité, positionnées à des endroits stratégiques sur l'aéronef, servant de balises au système de vision. Il serait ainsi vraisemblablement plus facile et plus rapide de repérer ces cibles. De plus, la probabilité de non-détection de la cible se trouverait améliorée.

En terminant, l'utilisation d'un système d'acquisition plus complet pour le banc d'essais est conseillée. En effet, l'emploi de l'ordinateur de bord requiert un temps de développement considérable pour le programme. De plus, la maintenance et les améliorations à apporter à ce système demandent un programmeur qualifié et de longues étapes de familiarisation. C'est pourquoi l'utilisation d'un ordinateur Pc muni d'une carte d'acquisition est recommandée. De cette façon il sera possible de programmer celui-ci

directement avec Labview. Le temps de développement sera ainsi accéléré et une qualification minimum sera requise pour la maintenance de ce système.

## BIBLIOGRAPHIE

- [1] Buchanan, J. (1990). *CMOS/TTL Digital Systems Design*. New York: McGraw-Hill.
- [2] Laplante, P.A.. (1997). *Real-Time Systems Design and Analysis*. Piscataway, N.J.: IEEE Press.
- [3] Harman T.L. (1991). *The Motorola MC68332 Microcontroller*. New Jersey, Prentice Hall.
- [4] Lafore R. (1995). *Object-Oriented Programming in C++*. Californie, Waite Group Press.
- [5] Joncas, S. (2000). *Conception et fabrication d'une aile rotative basse vitesse pour un hélicoptère à propulsion humaine*. Mémoire de maîtrise en génie mécanique. École de Technologie Supérieure.
- [6] Belleau, C. (2000). *Conception d'un système de contrôle pour un hélicoptère à propulsion humaine*. Mémoire de maîtrise en génie mécanique. École de Technologie Supérieure.
- [7] Michaud, C. (2000). *Conception d'un hélicoptère à propulsion humaine*. Mémoire de maîtrise en génie mécanique. École de Technologie Supérieure.



- [8] Boland, J-F. (1998). *Conception d'un ordinateur de bord pour un sous-marin à propulsion humaine*. Projet synthèse en génie électrique. École de Technologie Supérieure.
- [9] (1999) The Pressure Strain and Force Handbook, OMEGA
- [10] Site web de l'International Human Powered Vehicle Association  
<http://www.ihpva.org/>
- [11] E Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.
- [12] R. Lepage, *SYS-863 Vision par ordinateur: Notes de cours, École de technologie supérieure*, rédaction novembre 1997.

## **ANNEXE A**

### **Règlements du défi Igor Sikorsky**

## **1. General**

1.1 The prize will be awarded by the AHS to the entrant who first fulfils the conditions.

1.2 Additionally, an attempt will be registered with the Federation Aeronautique Internationale (FAI) as a World Record for Human-Powered Helicopter Flight duration.

## **2. Prize**

2.1 The AHS prize in U.S. \$20,000.

## **3. Eligibility**

3.1 The competition is international and is open to individuals or teams from any part of the world.

3.2 Any questions regarding the acceptance of entries, eligibility of entrant, pilot, crew or aircraft under these regulations, or any other matter relating to the AHS prize, the decision of the AHS is final.

3.3 All questions regarding the world record attempt will be governed by the sporting code of the FAI and rest exclusively with the NAC.

## **4. Conditions of Entry**

### **4.1 Aircraft**

4.1.1 The machine shall be a heavier-than-air machine. The use of lighter-than-air gases shall be prohibited.

4.1.2 The machine shall be a rotary wing configuration capable of vertical takeoff and landing in still air, and at least one member of the crew shall be non-rotating.

4.1.3 The machine shall be powered and controlled by the crew during the entire flight, including accelerating the rotor up to takeoff speed.

4.1.4 No devices for storing energy either for takeoff or for use in flight shall be permitted. Rotating aerodynamic components, such as rotor blades, used for lift and/or control are exempt from consideration as energy storing devices.

4.1.5 No part of the machine shall be jettisoned during the flight including the rotor spin-up and takeoff.

### **4.2 Crew**

4.2.1 The crew shall be those persons in the machine during takeoff and flight, and there shall be no limit set to their number.

4.2.2 No member of the crew shall be permitted to leave or enter the aircraft at any time during takeoff or flight.

4.2.3 No drugs or stimulants shall be used by any member of the crew. An assurance must be given to the official observers at the time of the attempt that this requirement has been met.

4.2.4 Up to two handlers or ground crew shall be permitted to assist in stabilizing the machine during takeoff and landing, but in such a manner that they do not assist in accelerating or decelerating any part of the machine.

## **4.3 Ground Conditions**

4.3.1 All attempts, which shall include the takeoff, shall be made over approximately level ground (i.e., with a slope not exceeding 1 in 100 in any direction).

4.3.2 All attempts shall be made in still air, which shall be defined as a wind not exceeding a mean speed of approximately one meter per second (3.1 kilometres per hour, 2.23T statute miles per hour, 1.5 nautical miles per hour) over the period of the flight.

## **4.4 Flight Requirements**

4.4.1 The flight requirements shall consist of hovering for one minute while maintaining flight within a 10-meter square. During this time the lowest part of the machine shall exceed momentarily 3 meters above the ground.

4.4.2 The machine shall be in continuous flight from takeoff to landing, and at no time during the flight shall any part of the machine touch the ground.

4.4.3 A reference point on the non-rotating part of the machine will be established as a means whereby the observers can judge that the machine stayed within the confines of the 10-meter square.

4.4.4 The one minute hovering time and the momentary achievement of 3 meters altitude is required to win the AHS prize. (However, the FAI 1980 regulations specify that only the duration of the flight and a momentary achievement of 3 meters altitude will be recorded for the FAX world record attempt, making it possible to achieve a world record without satisfying the AHS prize requirements.)

#### **4.5 Observation**

Every attempt shall be observed by the NAC or by any persons authorized by them to act as observers. It may take place in the competitors own country if it is affiliated to the FAI. In a country not so, it could be advantageous to conduct the flight in a neighboring country which is so affiliated.

#### **5. Applications for Entry**

5.1 Entry forms shall be obtained from and returned to the American Helicopter Society. 217 N. Washington St., Alexandria, VA 22314, (703) 684-6777. mail: ahs@ahs.org

5.2 The entry fee shall be U.. \$15 (made payable to the American Helicopter Society).

5.3 Each entry form shall contain an application for official observation of the competitors attempt.

5.4 The entrant shall undertake to abide by the conditions for official observation as set out on the entry form and application for official observation and shall undertake to defray all expenses incurred in connection with the official observation of the attempt.

5.5 The following fees and charges are made by the NAA for record attempts in Class I, Human Powered Aircraft. All attempts shall be for national and international records.

5.6 Final notice of the proposed time and place of the attempt requiring official observation may, if so desired, be sent to the AHS later than the entry form. It must in all cases be received at least thirty days before the proposed date of the attempt. This time is required by the NAC (the NAA in the U.S.A.) to

arrange for official observation. Applications will be considered in order of receipt.

5.7 Membership in the appropriate NAC and an FAI Sporting License is required for all crew members taking part in this competition. Application forms may be obtained from the NAC or the AHS. For this competition, a pilot's license is not required.

#### **6. General Conditions**

##### **6.1 Insurance**

The entrant must take out on behalf of himself, his crew, representatives or employees, liability insurance in such form and amount to be specified by the AHS, to indemnify the American Helicopter Society, the NAC and the FAI against any claims. Evidence that such insurance has been effected must be submitted with the application for official observation.

##### **6.2 Revision of Regulations**

6.2.1 These regulations shall remain in force until such time as the AHS considers it necessary to amend them, or the prize has been won.

6.2.2 The AHS reserves the right to add to, amend or omit any of these regulations and to issue supplementary regulations.

##### **6.3 Interpretation of Regulations**

The interpretation of these regulations or any of the regulations hereafter issued rest entirely with the AHS. The entrant shall be solely responsible to the official observer for due observance of these regulations and shall be the person with whom the official observers will deal in respect thereof, or any other question arising out of this competition.

## **ANNEXE B**

**Code source programme Héli-Tracker**

## ***Routine heli\_main***

function a = heli\_main(point\_depart, point\_arrivee,  
temps\_de\_simulation, noise\_density, backgroundX, backgroundY, blink,  
target\_intensity )

% Déclaration des variables Globales "

```
global cible_debut;           % Position XYZ initiale de la cible
global cible_fin;            % Position XYZ finale de la cible
global dt;                   % Temps total pour le déplacement
global blinkX_start;         % Début du blink, caméra X (secondes)
global blinkX_end;           % Fin du blink, caméra X (secondes)
global blinkY_start;         % Début du blink, caméra Y (secondes)
global blinkY_end;           % Fin du blink, caméra Y (secondes)
global cam
global subplotx;              % variables globales pour les options de
do_report_option              %
global subploty;
global subplot_cmpt;
global display_result;
```

if( nargin<8 ),

```
    error('Erreur: Necessite au moins 4 arguments. Prototype:
heli_main(point_depart, point_arrivee, temps_de_simulation,
noise_density, image_fond_X, image_fond_Y, blink, target_intensity )');
end
```

% Initialisation des variables globales

% Les coordonnées de fin doivent être plus grande que les coordonnées de début.

```
cible_debut = point_depart; % Coordonnees XYZ de début de la cible
                             % (pixels)
cible_fin = point_arrivee;   % Coordonnées XYZ de fin de la
                             % cible (pixels)
dt = temps_de_simulation;    % Temps total de déplacement en
                             % secondes
noise = noise_density;       % Densité du bruit "poivre & sel"
blinkX_start = 0;            % 5Interval de blink pour la caméra X
blinkX_end = 10;             % 7--> Mettre à zéro pour désactiver le blink
blinkY_start = 0;            % 1 Interval de blink pour la caméra Y
blinkY_end = 10;             % 8 --> Mettre à zéro pour désactiver le blink
```

```
subplotx = 6;                %Figure 6 * 4
subploty = 4;
display_result = 0; %par défaut, on n'affiche rien
found = [0 0];
```

```
subplot_cmpt = [1 1];
do_report_option = 1;
```

%Variables locales definissant les contraintes de detections de la cible

%Note Indice 0=camX 1=camY

%Image de la Camera en X

```
min_cam_area = [100 100]; %Aire minimum de la cible indice
```

```

max_cam_area          = [330 330]; %Aire maximum de la cible
max_cam_delta_xy      = [200 200]; %Déplacement maximum en X ou Y entre
% détections d'images
max_cam_delta_z       = [150 150]; %Déplacement
% maximum en Z entre 2 détections d'images

%fermeture de toutes les figures en cours
close all;

display_result = 0 ;
if( display_result )
    figure(1);
end

%initialisation de la fonction heli_control
[fov, target_pos] = heli_control;
testvar = fov;

% la boucle de simulation va de 0 a delta t seconde par increment de 1

if(do_report_option)
    profile on
end

for temps=0:dt

    [im_sim(:,:,,1), im_sim(:,:,,2)] =
heli_sim(temps,backgroundX,backgroundY,noise,blink, target_intensity); %
Génère une nouvelle image

    for cam = 1:2

        display_result = 1;
        if( display_result )
            %figure(1);%Make sure it is not figure 1
            %figure(2);%Make sure it is not figure 2
            %figure,
            imshow(im_sim(:,:,,cam)),axis xy;
            title(sprintf('init t=%d',temps));
            figure(cam);
            subplot(subplotx, subploty, subplot_cmpt(cam) );
            imshow(im_sim(:,:,,cam)),axis xy;
            C=[fov(cam,1),fov(cam,2);...
                fov(cam,1),fov(cam,4); ...
                fov(cam,3),fov(cam,4); ...
                fov(cam,3),fov(cam,2); ...
                fov(cam,1),fov(cam,2)];
            track_line(cam)=line(C(:,2),C(:,1));
            title(sprintf('init t=%d',temps));
            subplot_cmpt(cam) = subplot_cmpt(cam) +1 ;
        end

        display_result = 0;
        im_fov = heli_fov(im_sim(:,:,,cam),fov(cam,:));

        img_bw = heli_color_to_bw(im_fov,'blue',50);
    end
end

```

```

    %Affichage de l'image initiale
    display_result = 0;
    if( display_result )
        figure,
        imshow(img_bw),axis xy;
        title(sprintf('cam=%d t=%d noir/blanc',cam,temps));
        figure(cam);
        subplot(subplotx, subploty, subplot_cmpt(cam) );
        imshow(img_bw), axis xy;
        title(sprintf('cam=%d t=%d noir/blanc',cam,temps));
        subplot_cmpt(cam) = subplot_cmpt(cam) +1 ;
    end

    display_result = 0;
    img_bw_flt = heli_bw(img_bw, 100);

    %Affichage de l'image filree binaire
    display_result = 0;
    if( display_result )
        figure,
        imshow(img_bw_flt),axis xy;
        title(sprintf('cam=%d t=%d binaire',cam,temps));
        figure(cam);
        subplot(subplotx, subploty, subplot_cmpt(cam) );
        imshow(img_bw_flt),axis xy;
        title(sprintf('cam=%d t=%d binaire',cam,temps));
        subplot_cmpt(cam) = subplot_cmpt(cam) +1 ;
        disp(sprintf('Recherche de la cible sur l'image de la camera
%d pour t=%d ',cam,temps));
    end

    display_result = 0;
    [found(cam), last_cam_pos_xy(cam), last_cam_pos_z(cam)] = ...
        heli_detect(img_bw_flt, target_pos(cam,2)-fov(cam,2),
target_pos(cam,1)-fov(cam,1), ...
        min_cam_area(cam), max_cam_area(cam), max_cam_delta_xy(cam),
max_cam_delta_z(cam) );

    target_detect(cam,:) = [last_cam_pos_z(cam), last_cam_pos_xy(cam)
];

end %for camera

[fov, target_pos] = heli_control(found, target_detect);

display_result =1 ;
if( display_result)
    for cam = 1:2
        if( found(cam) )
            figure(cam);
            subplot(subplotx, subploty, subplot_cmpt(cam)-1);
            set(track_line(cam), 'Color','red');

            C=[target_pos(cam,2)-20,target_pos(cam,1)-20;...
target_pos(cam,2)+20,target_pos(cam,1)+20];
            track_line(cam)=line(C(:,1),C(:,2));

```





```

skipX = 0;
skipY = 0;

coul_cible = [0 0 target_intensity];           % Niveau de
couleur RGB de la cible

fprintf('intensite = %d',coul_cible(3));

% *****
% Déplacement de la cible
% *****
% Si le temps t est > que dt, on sature a dt
if t >= dt
    t = dt;
end

% Vérifie si on est dans l'interval de blink pour l'image X
if t >= blinkX_start & t <= blinkX_end & blink == 1
    skipX = 1;
end

% Vérifie si on est dans l'interval de blink pour l'image Y
if t >= blinkY_start & t <= blinkY_end & blink == 1
    skipY = 1;
end

% Calcul du déplacement total XYZ en pixels
dx = cible_fin(1) - cible_debut(1);
dy = cible_fin(2) - cible_debut(2);
dz = cible_fin(3) - cible_debut(3);

% Calcul du nombre de pixel / seconde
pas_x = dx / dt;
pas_y = dy / dt;
pas_z = dz / dt;

% Calcul du déplacement XYZ en pixel
dep_x = round(pas_x * t);
dep_y = round(pas_y * t);
dep_z = round(pas_z * t);

% Calcul du offset pour obtenir le coin inferieur gauche de la cible
n2 = round(n/2); % Hauteur divise par 2
m2 = round(m/2); % Largeur divise par 2

if skipX ~= 1
    imX(cible_debut(3)-n2+dep_z:cible_debut(3)-
n2+dep_z+n,cible_debut(1)-m2+dep_x:cible_debut(1)-m2+dep_x+m,1) =
coul_cible(1);
    imX(cible_debut(3)-n2+dep_z:cible_debut(3)-
n2+dep_z+n,cible_debut(1)-m2+dep_x:cible_debut(1)-m2+dep_x+m,2) =
coul_cible(2);
    imX(cible_debut(3)-n2+dep_z:cible_debut(3)-
n2+dep_z+n,cible_debut(1)-m2+dep_x:cible_debut(1)-m2+dep_x+m,3) =
coul_cible(3);
end

```

```

if skipY ~= 1
    imY(cible_debut(3)-n2+dep_z:cible_debut(3)-
n2+dep_z+n,cible_debut(2)-m2+dep_y:cible_debut(2)-m2+dep_y+m,1) =
coul_cible(1);
    imY(cible_debut(3)-n2+dep_z:cible_debut(3)-
n2+dep_z+n,cible_debut(2)-m2+dep_y:cible_debut(2)-m2+dep_y+m,2) =
coul_cible(2);
    imY(cible_debut(3)-n2+dep_z:cible_debut(3)-
n2+dep_z+n,cible_debut(2)-m2+dep_y:cible_debut(2)-m2+dep_y+m,3) =
coul_cible(3);
end

% Ajoute du bruit dans l'image
if noise > 0
    imX = imnoise(imX,'salt & pepper',noise);
    imY = imnoise(imY,'salt & pepper',noise);
end

```

### ***Programme heli\_fov***

```

function im_out = heli_fov(im_in,fov)
im_out = im_in(fov(1):fov(3),fov(2):fov(4),:);

```

### ***Programme heli\_color\_to\_bw***

```

function image_retour = heli_color_to_bw(image, color, seuil)
%HELI_COLOR_TO_BW( image, [color] )
%Ce script convertit une image RGB(couleur) en noir et blanc.
%Ce script permet d'extraire seulement une couleur de l'image en
specifiant le
%parametre color a une des valeurs suivantes
% 'red' pour extraire seulement la composante rouge de l'image
% 'green' pour extraire seulement la composante verte de l'image
% 'blue' pour extraire seulement la composante bleue de l'image
% La valeur de seuil permet d'enlever (mettre a zero) de cette nouvelle
image tous les
% pixels des autres couleurs qui sont plus haut que cette valeur de
seuil
%
if( nargin<3 ),
    error('Erreur: Necessite au moins 3 arguments. Prototype:
heli_color_to_bw(image, color, seuil)');
end
color = [color, ' '];
code = lower(color(1:2));

if( code == 're' ), %rouge
    image_retour = (image(:,:,2)<seuil); %Gardes tous les pixels vert
plus bas que le seuil
    image(:,:,1) = double(image(:,:,1)) .* double(image_retour);
%enleves les pixels plus haut que le seuil de la composante choisie
    image_retour = image(:,:,3) <seuil; %Gardes tous les pixels bleu
plus bas que le seuil
    image(:,:,1) = double(image(:,:,1)) .* double(image_retour);
%enleves les pixels plus haut que le seuil de la composante choisie

```

```

    image_retour = uint8(image(:,:,1)); %on garde seulement le rouge
elseif( code == 'gr' ), %vert
    image_retour = (image(:,:,1)<seuil); %Gardes tous les pixels rouge
    plus bas que le seuil
    image(:,:,2) = double(image(:,:,2)) .* double(image_retour);
    %enleves les pixels plus haut que le seuil de la composante choisie
    image_retour = image(:,:,3) <seuil; %Gardes tous les pixels bleu
    plus bas que le seuil
    image(:,:,2) = double(image(:,:,2)) .* double(image_retour);
    %enleves les pixels plus haut que le seuil de la composante choisie
    image_retour = uint8(image(:,:,2)); %on garde seulement le vert
else %bleu , default
    image_retour = (image(:,:,1)<seuil); %Gardes tous les pixels rouge
    plus bas que le seuil
    image(:,:,3) = double(image(:,:,3)) .* double(image_retour);
    %enleves les pixels plus haut que le seuil de la composante choisie
    image_retour = image(:,:,2) <seuil; %Gardes tous les pixels vert
    plus bas que le seuil
    image(:,:,3) = double(image(:,:,3)) .* double(image_retour);
    %enleves les pixels plus haut que le seuil de la composante choisie
    image_retour = uint8(image(:,:,3)); %on garde seulement le bleu
end

```

### ***Programme heli\_bw***

```

function image_retour = heli_bw(image, seuil)
%HELI_BW( image )
%Ce script filtre l'image d'entre pour y eliminer les bruits due a la
capture d'image
global subplotx;
global subploty;
global subplot_cmpt;
global display_result;
global cam;

if( nargin<2 ),
    error('Erreur: Necessite au moins 2 arguments. Prototype:
heli_bw(image, seuil)');
end

%On commence par augmenter le contraste avec un scaling (Formule: Pixel
= (pixel - min) * 255 / max
max_value = double(max(max(image)))/255;
min_value = seuil/255; %Valeur de seuil
image = imadjust(image,[min_value ,max_value],[0,1]);
if( display_result )
    figure,
    imshow(image),axis xy;
    title(sprintf('Augmentation du contraste'));
    figure(cam);
    subplot(subplotx, subploty, subplot_cmpt(cam) );
    imshow(image),axis xy;
    title(sprintf('Augmentation du contraste'));
    subplot_cmpt(cam) = subplot_cmpt(cam) +1 ;
end

```

```

image = medfilt2(image,[4,4]);

if( display_result )
    figure,
    imshow(image),axis xy;
    title(sprintf('Median 4*4'));
    figure(cam);
    subplot(subplotx, subploty, subplot_cmpt(cam) );
    imshow(image),axis xy;
    title(sprintf('Median 4*4'));
    subplot_cmpt(cam) = subplot_cmpt(cam) +1 ;
end

image = uint8(heli_bin(image, 35));
if( display_result )
    figure,
    imshow(image),axis xy;
    title(sprintf('Image binarisee'));
    figure(cam);
    subplot(subplotx, subploty, subplot_cmpt(cam) );
    imshow(image),axis xy;
    title(sprintf('Image binarisee'));
    subplot_cmpt(cam) = subplot_cmpt(cam) +1 ;
end

image_retour = image;

```

### ***Programme heli\_detect***

```

function [found,target_pos_x, target_pos_y ] = heli_detect(image,
last_x, last_y, min_area, max_area, max_delta_x, max_delta_y )
%HELI_BW( image )
%Ce script detecte la cible dans l'image d'entree

global subplotx;
global subploty;
global subplot_cmpt;
global display_result;
global cam;

persistent last_found;

if( nargin<3 ),
    error('Erreur: Necessite au moins 1 arguments. Prototype:
heli_detect(image, aire_min aire_max )');
end

found = 0;

[L,NUM] = bwlabel(image,8);

if(display_result)
    disp(sprintf('%d objets detectes',NUM));
end

if( NUM )
    for i = 1:NUM

```

```

[r,c] = find(L == i);
x1=min(c);
x2=max(c);
y1=min(r);
y2=max(r);
pos_x = round((x1 + x2) / 2);
pos_y = round((y1 + y2) / 2);

image_object = bwselect(image,c,r,3);
area = round(bwarea(image_object));
%The previous two lines can be replace the following, PGP
area = size(r);
area = area(1); %r is a X by 1 vector, don't care about the 1

if(display_result & 0)
    disp(sprintf('area = %d ',area));
end

if( area < min_area | area > max_area )
    disp(sprintf('objet %d rejete, aire trop petite ou trop
grande ',i));
    disp(sprintf('aire = %d ',area));
elseif( abs(pos_x - last_x) > max_delta_x | abs(pos_y - last_y) >
max_delta_y )
    disp(sprintf('objet %d rejete, deplacement hors des limites
permises',i));
    disp(sprintf('x : %d y : %d ',pos_x,pos_y));
else
    found = 1;
    target_pos_x = pos_x;
    target_pos_y = pos_y;
    if( display_result)
        disp(sprintf('object %d : ',i));
        disp(sprintf('area = %d ',area));
        disp(sprintf('heli_detect: x:%d y:%d', pos_x,pos_y));
        figure,
        imshow(image_object),axis xy;
        title(sprintf('Objet %d',i));
        figure,
        imshow(bwperim(image_object)),axis xy;
        title(sprintf('Perimetre Objet %d',i));
        figure(cam);
        subplot(subplotx, subploty, subplot_cmpt(cam) );
        imshow(image_object),axis xy;
        title(sprintf('Objet %d',i));
        subplot_cmpt(cam) = subplot_cmpt(cam) +1 ;
        subplot(subplotx, subploty, subplot_cmpt(cam) );
        imshow(bwperim(image_object)),axis xy;
        title(sprintf('Perimetre Objet %d',i));
        subplot_cmpt(cam) = subplot_cmpt(cam) +1 ;
    end
end
end %for loop
else
    if( display_result )
        disp(sprintf('Aucun objet detecte'));
    end
end

```

```

        end
    end

    if( found == 0)
        target_pos_x = last_x;
        target_pos_y = last_y;
    end
end

```

### ***Programme heli\_control***

```

heli_control.m
%
%Calcul la position en pixel de la cible de chacune des images et
%evalue
%le champ de vision pour le prochain traitement.
%
%Initialisation de la fonction:
% [champs_de_vision, position_cible] = heli_control
%
%Calcul du champ de vision et le la position des cibles:
% [champs_de_vision, position_cible] = heli_control [status, detect]
% - status:  etat des cibles
% - detect:  position cibles detectees
%
%Jean Denis Belanger
%Jean-Francois Boland
%Pierre Grondin
%avril 2000

function [fov_out, target_pos_out] = heli_control (varargin);

persistent GOOD BAD ;
persistent good_count bad_count;
persistent fov_index_max fov_index;
persistent fov_size_table;
persistent image_limit_min image_limit_max;
persistent fov target_pos;

%Initialisation de la fonction
%[fov_out, target_pos_out] = heli_control
%if nargin == 0
%    heli_control_init;

%Mise a jour du champ de vision et de la position de la cible
%[fov_out, target_pos_out] = heli_control(track_status,
target_detect_pos)
elseif nargin == 2
    track_status = varargin{1};
    target_detect_pos = varargin{2};

    for cam = 1:2

        if track_status(cam) == 0
            heli_control_bad_track;
        end
    end
end

```

```

        if track_status(cam) == 1
            heli_control_good_track;
            target_pos(cam,:) = fov(cam,1:2) + target_detect_pos(cam,:) -
1;
        end

        heli_control_set_fov;

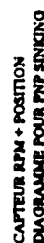
    end %for cam = 1:2
end % if/elseif

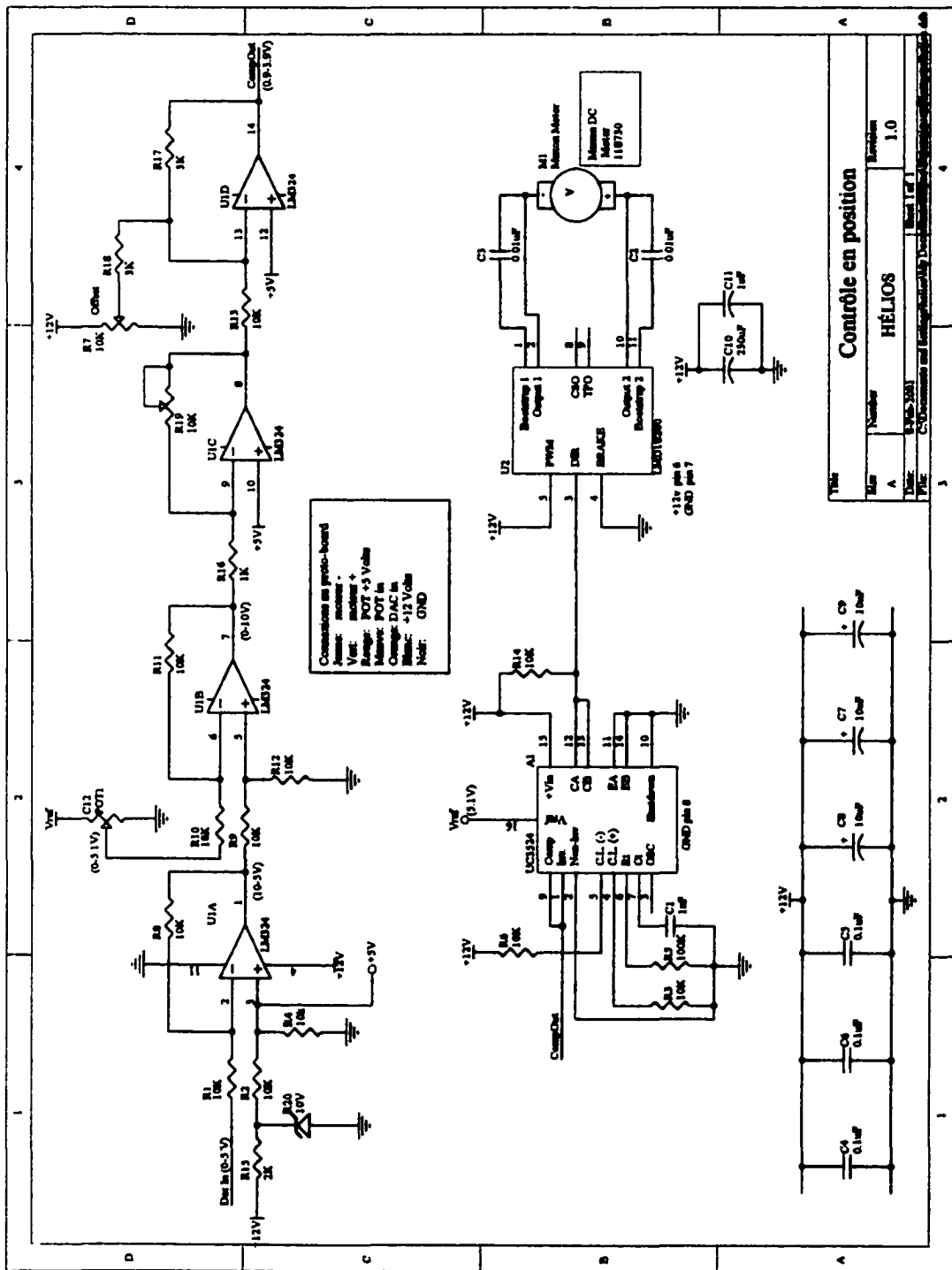
fov_out = fov;
target_pos_out = target_pos;

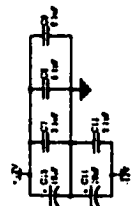
```



**ANNEXE C**  
**Schémas électriques**





[illegible]